1993年1月1日発行（毎月1回1日発行）第12巻1号通巻129号 昭和58年11月2日第三種郵便物認可

PERSONAL COMPUTER MAGAZINE for MZ, X1, and X68000

# Oh!X

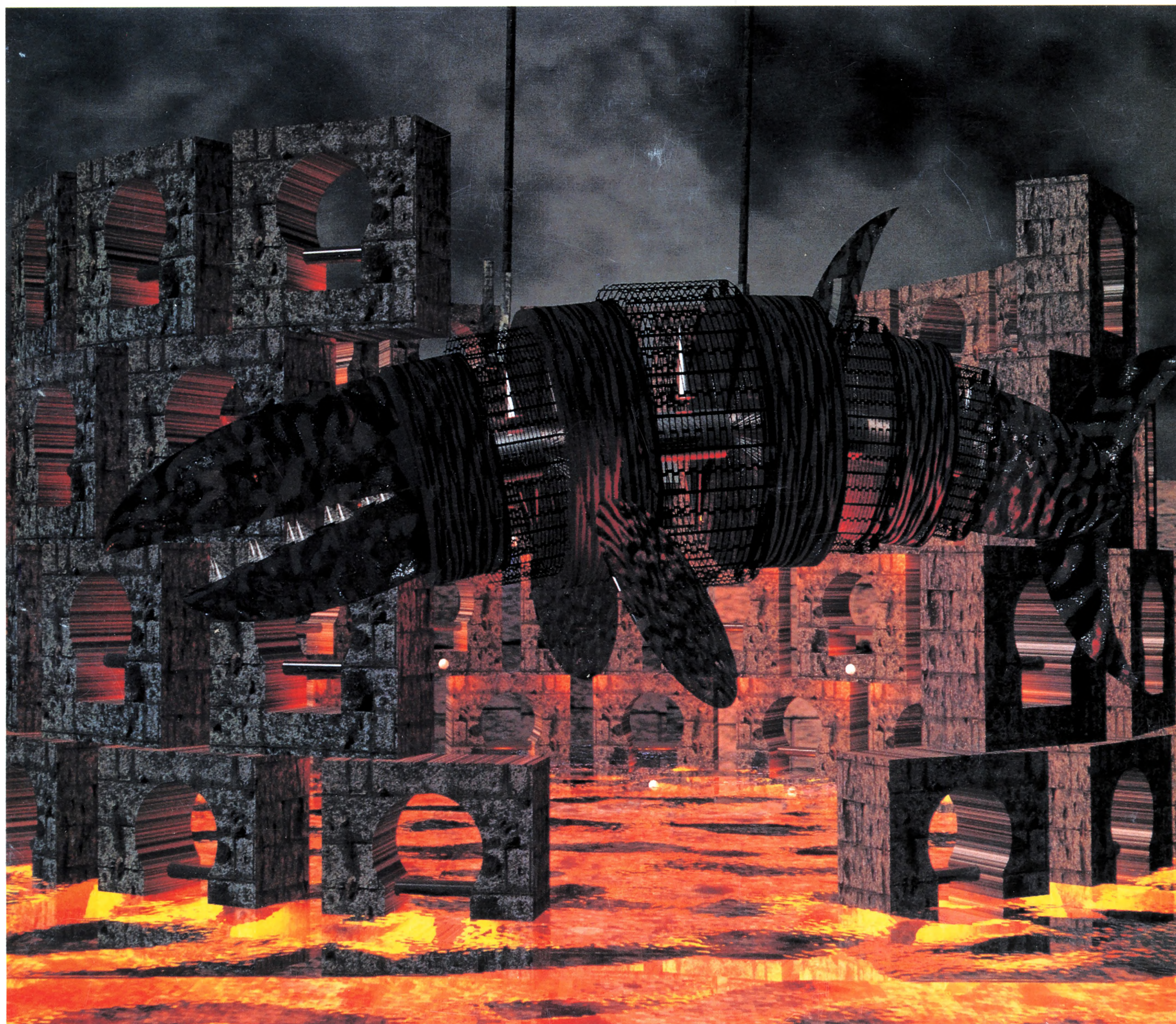
## 特集 D.I.Y.ハードウェア

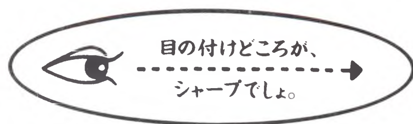
X68000用アクセラレータを作る/バーコードリーダーの製作

新製品紹介 サブMPUボード POLYPHON/THUNDER WORD SOUND SX-68K/CD-ROM SX広辞苑/カードゲーム GOLF

1
1993

SOFT BANK オー・エックス
定価600円






目の付けどころが、
シャープでしょ。

# “感性”咲かせるワー

## POWER WORKSTATION

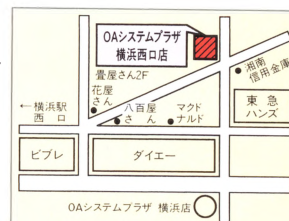
インテリジェントなパフォーマンスを誇るX68000Compact XVIと多彩にラインアップされたペリフェラル。感性を刺激するクリエイティブなワークステーション環境が自在に構築できます。

- パーソナルワークステーション(2HD3.5インチFDDタイプ・本体＋キーボード＋マウス)
  **CZ-674C-H**(グレー) 標準価格**298,000**円(税別)
- 15型カラーディスプレイテレビ
  **CZ-614D-TN**(チタンブラック)・**-BK**(ブラック) 標準価格**135,000**円(税別)
  ■ディスプレイテレビ/CZ-6TU用RGBケーブル**CZ-6CR1** 標準価格**4,500**円(税別)
  ■ディスプレイテレビ/CZ-6TU用TVコントロールケーブル**CZ-6CT1** 標準価格**5,500**円(税別)
- 80MB内蔵用ハードディスクドライブ
  **CZ-68HA** 好評発売中
- 5.25インチ増設用フロッピーディスクドライブ
  **CZ-6FD5** 標準価格**99,800**円(税別・接続ケーブル同梱)
- 光磁気ディスクユニット
  **CZ-6MO1** 標準価格**450,000**円(税別)
  ■SCSI変換ケーブル**CZ-6CS1** 標準価格**12,000**円(税別)
- 2MB増設RAMボード
  **CZ-6BE2D** 標準価格**54,800**円(税別・取り付け費別)
  ■2MB増設RAM**CZ-6BE2B** 標準価格**54,800**円(税別・取り付け費別)×2
  ■数値演算プロセッサ**CZ-6BP2** 標準価格**45,800**円(税別・取り付け費別)
- 48ドット熱転写カラー漢字プリンタ
  **CZ-8PC5-BK**(ブラック) 標準価格**96,800**円(税別)
- MIDIボード
  **CZ-6BM1A** 標準価格**26,800**円(税別)
- インテリジェントコントローラ
  **CZ-8NJ2** 標準価格**23,800**円(税別)




## X68000 見・体・験フェア 開催

開催日時：12月19日(土)・20日(日)
10:00～19:00
会　場：(株)OAシステムプラザ横浜西口店
横浜市西区南幸2-8-12 松山ビル2F
☎(045)314-6637
交通／JR他横浜駅西口すぐ
■主催：(株)OAシステムプラザ
■問い合わせ先：シャープエレクトロニクス販売(株)
横浜情報営業部 ☎(045)753-5585 担当・沢井
★ゲーム大会、来場者記念品あり




資料請求券
X68000ペリフェラル
Oh!X
1係

X68000
PERSONAL WORKSTATION・XVI
Compact

# クステーション環境。

## GRAPHIC WORKSTATION

- パーソナルワークステーション(2HD3.5インチFDDタイプ・本体＋キーボード＋マウス)
  **CZ-674C-H**(グレー) 標準価格**298,000**円(税別)
- 21型カラーディスプレイ **CU-21HD** 標準価格**148,000**円(税別)
- 80MB内蔵用ハードディスクドライブ **CZ-68HA** 好評発売中
- 光磁気ディスクユニット **CZ-6MO1** 標準価格**450,000**円(税別)
  ■SCSI変換ケーブル**CZ-6CS1** 標準価格**12,000**円(税別)
- 2MB増設RAMボード **CZ-6BE2D** 標準価格**54,800**円(税別・取り付け費別)
  ■2MB増設RAM**CZ-6BE2B** 標準価格**54,800**円(税別・取り付け費別)×2
  ■数値演算プロセッサ**CZ-6BP2** 標準価格**45,800**円(税別・取り付け費別)
- カラーイメージスキャナ
  **CZ-8NS1** 標準価格**188,000**円(税別)
  ■スキャナ用パラレルボード**CZ-6BN1** 標準価格**29,800**円(税別)
- カラーイメージジェット
  **IO-735X-B**(ブラック) 標準価格**248,000**円(税別)
  ■接続ケーブル**IO-73CX** 標準価格**5,500**円(税別)




## STANDARD WORKSTATION

- パーソナルワークステーション
  (2HD3.5インチFDDタイプ・本体＋キーボード＋マウス)**CZ-674C-H**(グレー) 標準価格**298,000**円(税別)
- 14型カラーディスプレイ**CZ-608D-H**(グレー) 標準価格**94,800**円(税別)
- 5.25インチ増設用フロッピーディスクドライブ**CZ-6FD5** 標準価格**99,800**円(税別・接続ケーブル同梱)



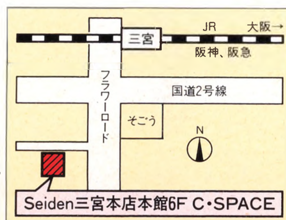

## TFT COLOR LCD WORKSTATION

- パーソナルワークステーション
  (2HD3.5インチFDDタイプ・本体＋キーボード＋マウス)**CZ-674C-H**(グレー) 標準価格**298,000**円(税別)
- 10.4型カラー液晶ディスプレイ**LC-10C1-H**(グレー) 標準価格**598,000**円(税別)
  ■接続ケーブル**AN-1515X** 標準価格**4,200**円(税別)

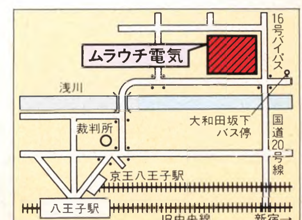
※カラー液晶ディスプレイを接続してご使用の場合、SX-WINDOW上のアプリケーション利用に限定されます。

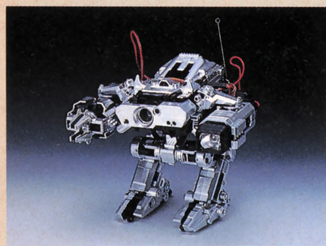



開催日時:**12月19日(土)・20日(日)**
11:00～17:00
会　場:**Seiden三宮本店本館6F C・SPACE**
神戸市中央区三宮町1丁目5-8
☎(078)391-8171
交通/JR、阪神、阪急三宮駅下車徒歩5分
■主催:せいでん三宮本店
■問い合わせ先:シャープライブエレクトロニクス販売㈱ 近畿統轄神戸支店☎(078)431-4361 担当・小田
★来場者記念品あり

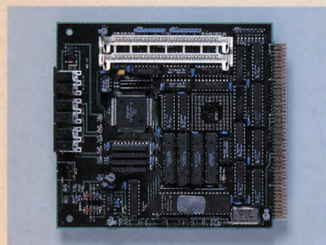



開催日時:**12月23日(水)**
10:00～19:00
会　場:**㈱ムラウチ**
東京都八王子市大和田町5-1-21
☎(0426)42-6211
交通/JR八王子から日野駅行きバス大和田坂下 下車
■主催:㈱ムラウチ
■問い合わせ先:シャープライブエレクトロニクス販売㈱ 三多摩支店☎(0425)84-0674 担当・藤井
★山下章氏来場＋ゲーム大会(参加記念品有)





特集　D.I.Y. ハードウェア



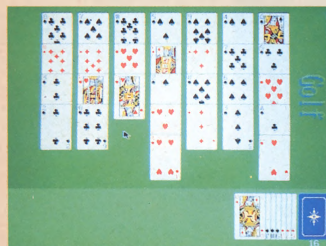
POLYPHON



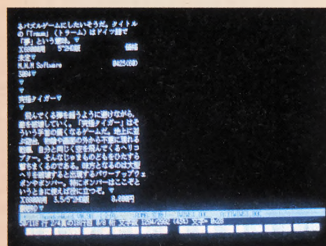

オーバーテイク




ストライダー飛竜




カードゲーム GOLF



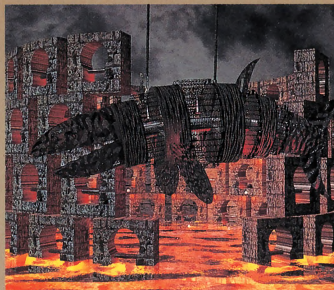
THUNDER WORD

# Oh!X

# C O N T




〈スタッフ〉
●編集長／前田　徹　●副編集長／植木章夫　●編集／浅井研二　山田純二　豊浦史子　●協力／有田隆也　中森　章　林　一樹　吉田幸一　華門真人　吉田賢司　影山裕昭　大和　哲　村田敏幸　丹　明彦　三沢和彦　長沢淳博　宮島　靖　金子俊一　浦川博之　石上達也　柴田　淳　御木徳高　瀧　康史　●カメラ／杉山和美　●イラスト／山田晴久　寺尾響子　高橋哲史　川原由唯　●アートディレクター／島村勝頼　●レイアウト／元木昌子　ADGREEN　●校正／グループごじら



表紙絵：塚田　哲也

# E　N　T　S

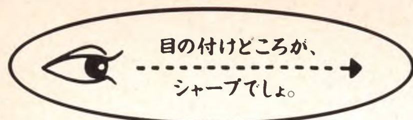
1993 JAN.
1





■広告目次

SHARP




# X68000 CompactXVI NEWS

## Opinion 1

### (ハードディスクが使いたい。)

Compact専用の内蔵ハードディスクが登場しました。SCSI仕様の80MB。場所を取らずに高速・大容量ファイル環境を実現します。

■内蔵用ハードディスクドライブ(CZ-674C専用)
CZ-68HA……………………………好評発売中
※取りつけに関してはシャープお客様ご相談窓口にてご相談ください(取りつけ費別)。

さらに大容量をお望みの場合、外付け用のSCSI端子で一般のSCSIハードディスクも接続可能。フルピッチSCSI端子とハーフピッチSCSI端子を接続するためのSCSI変換ケーブルも用意しています。

■SCSI変換ケーブル
CZ-6CS1………標準価格12,000円(税別)


CZ-6CS1

## Opinion 2

### (従来のソフト資産を活かしたい。)

これについても、Compact専用の外付け5インチフロッピーディスクユニットを用意していますから、従来の68シリーズの資産を有効活用できます。3.5インチと5インチの間でのデータのやりとりも可能。また、CZ-674C及びCZ-6FD5のスイッチ設定を変えれば、5インチソフトからの起動が可能になり、市販ソフトなどそのまま使えます。



■増設用5インチ・フロッピーディスク・ユニット(CZ-674C専用)
CZ-6FD5……………………標準価格99,800円(税別)

## Opinion 3

### (ディスプレイテレビを接続したい。)

Compactは、従来のシリーズと比べ体積比44%と小さいため、コネクタの形状も異なっていますが、このケーブルを使用することにより、ディスプレイテレビやRGBシステムチューナーを利用できます。


CZ-6CR1


CZ-6CT1



■15型カラーディスプレイテレビ(スピーカー・チルトスタンド同梱)
CZ-614D-TN……………標準価格135,000円(税別)

■ディスプレイテレビ/CZ-6TU用RGBケーブル
CZ-6CR1……………………標準価格　4,500円(税別)

■ディスプレイテレビ/CZ-6TU用テレビコントロールケーブル
CZ-6CT1……………………標準価格　5,500円(税別)

資料請求券
X68000ペリフェラル
Oh!X
1係

# パーソナルワークステーション
# X68000 Compact XVIについての
# ご意見、ご要望にお応えします。

## Opinion 4

### （ メモリ環境をパワーアップしたい。 ）

Compactは2MBのメインメモリを標準装備していますが、本体内で最大8MBまで拡張できます。

| | 容量 | 周辺機器 |
|---|---|---|
| 標準 | 2MB | —— |
| 拡張 | 4MB | CZ-6BE2D |
| | 6MB | CZ-6BE2B |
| | 8MB | CZ-6BE2B×2 |

■2MB増設RAMボード CZ-6BE2D 標準価格54,800円（税別）

■2MB増設RAM CZ-6BE2B 標準価格54,800円（税別）

※取りつけに関してはシャープお客様ご相談窓口にてご相談ください（取りつけ費別）。

## Opinion 5

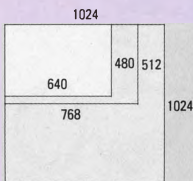
### （ 液晶ディスプレイとSX-WINDOWの関係は? ）

液晶ディスプレイ（LC-10C1-H 標準価格598,000円・税別）の解像度は640×480ドット。Compactでは、従来のX68000シリーズの画面モードにこの画面モードをプラス。解像度の制約を受けないウィンドウ環境ならではの機能です。このようにSX-WINDOW環境の確立により、ハードウェアに依存しない快適な操作環境が実現します。

SX-WINDOWの実画面エリア 1024×1024ドット

SX-WINDOWの通常表示エリア 768× 512ドット

SX-WINDOW上での液晶ディスプレイの表示エリア 640× 480ドット







## Opinion 6

### （数値演算プロセッサはほんとに速い?）

ご存じのようにMPU68000自体は複雑な計算（浮動小数点演算）を単純な計算の組み合わせで行っています。X68000シリーズに装備されている浮動小数点演算パッケージ「FLOAT2.X」は、よく使う単純な組み合わせをまとめたもの。数値演算プロセッサは、いわばこのパッケージの機能を、ハードウェアで高速に実現し、MPUの負担を軽くするものです。アプリケーションプログラムの中には浮動小数点演算を必要としないものもあるため、すべてのプログラムが高速になるわけではありませんが、レイトレーシングなど大量の実数演算を必要とするソフトウェアの場合、飛躍的な実行速度の向上が期待できます。

■数値演算プロセッサ CZ-6BP2 標準価格45,800円（税別）

※数値演算プロセッサはCZ-6BE2D上に装着します。

※取りつけに関してはシャープお客様ご相談窓口にてご相談ください（取りつけ費別）。



X68000
PERSONAL WORKSTATION・XVI
Compact

本体＋キーボード＋マウス
2HD3.5インチFDDタイプ CZ-674C-H（グレー） 標準価格298,000円（税別）
14型カラーディスプレイ（ドットピッチ0.28㎜）
CZ-608D-H（グレー） 標準価格94,800円（税別）

■お問い合わせは…電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06) 621-1221（大代表）
電子機器事業本部AVCシステム事業推進室 〒162 東京都新宿区市谷八幡町8番地 ☎(03)3260-1161（大代表）
シャープ株式会社

入力から、編集、出力、複写まで、カラーの世界は、シャープです。




# BLACK SPIRITS

## ▶INPUT

### X68000用パラレルインタフェースを標準装備した高速コンパクト型イメージスキャナ。

**カラーイメージスキャナ JX-220X……標準価格168,000円(税別)**

●A4サイズの原稿を約50秒※1で高速読み取り●CCDセンサー採用。さらに中間調処理でシャープでリアルな画像を再現●ディザパターン指定機能※2や濃度補正機能※2など高度な画像処理機能で緻密な読み取りが可能●解像度200ドット/インチ(約7.9ドット/㎜)。ズーム機能で1%きざみの拡大、縮小も可能●色ずれの少ない線順次(1走査)読み取り●X68000シリーズ用「スキャナツール」ソフトを標準装備●プリンタと直接接続することによりダイレクトプリント※3が可能●RS-232Cインタフェース/X68000シリーズ用専用パラレルインタフェースを標準装備。

※1:A4、2値出力、コンピュータへの実転送時間。
※2:表記機能はJX-220X本体使用であり、付属ユーティリティ使用時は異なります。
※3:別売のパラレルインタフェースケーブル(JX-22PC標準価格12,000円(税別)が必要です。



## ▶OUTPUT

### 3種類の制御コマンドモードを搭載。質感も鮮やかに再現する高品位カラーイメージジェット。

**カラーイメージジェット IO-735X-B……標準価格248,000円(税別)**

●シャープ独自のIOシリーズコマンド(Gモード)に加え、NM-9900モード(Nモード)、ESC/P24-84C準拠モード(Pモード)をサポート。一般文書の作成から、各種デザイン、建築用パースなどのCAD分野に対応●発色性に優れた普通紙対応の新黒インキ採用。専用紙はもちろんオフィスでよく使われる普通紙にも鮮明カラー印字●プリントバッファメモリ(128KB)の内蔵で、ホストコンピュータの拘束時間を軽減●48ノズル(各色12ノズル)採用の高速印字。A4-1ページを※約90秒でプリント(データ受信時間除く)●ビジネス用途に適したB4横用紙幅対応●OHPフィルム(専用)にも鮮明プリント●ノンインパクト方式ならではの静粛印字●インキ補充は簡単、経済的なカートリッジ方式

※261×174㎜領域





### IO-735X-B 対応アプリケーション

●SX-WINDOW対応ペイントツール
**Easypaint** SX-68K
CZ-263GW 標準価格12,800円(税別)

●WYSIWYGを実現、ドローグラフィックソフト
**CANVAS** PRO-68K
CZ-249GS 標準価格29,800円(税別)

●オリジナリティを活かせるポップアップツール
**NEW Printshop** PRO-68K ver2.0
CZ-221HS 標準価格20,000円(税別)

●マルチワープロ
**Multiword** PRO-68K
CZ-225BS 標準価格32,000円(税別)

●高速カード型リレーショナルデータベース
**CARD** PRO-68K ver2.0
CZ-253BS 標準価格29,800円(税別)

●パソコン通信もできるメモリ常駐型ソフト
**Teleportion** PRO-68K
CZ-258BS 標準価格22,800円(税別)

●これからの高速通信をサポート
**Communication** PRO-68K ver2.0
CZ-257CS 標準価格19,800円(税別)

資料のご請求・お問い合わせはコンシューマーセンター
●東日本相談室 〒261 千葉市美浜区中瀬1丁目9番2号 ☎(043)297-1221(大代表) ●西日本相談室 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)

# 待望のSX-WINDOW 開発支援ツール、登場。

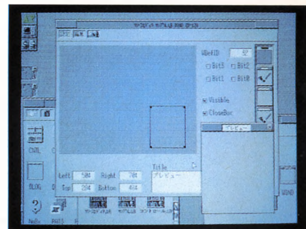
## SX-WINDOW 開発キット Workroom SX-68K

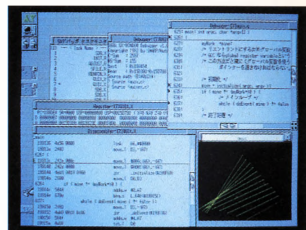
NEW

**CZ-288LWD 12月発売予定**

SX-WINDOW用のソフト開発に必要な開発ツールやサンプルプログラムを装備。プログラムの編集、リソースの作成、コンパイル、デバッグといった一連の作業をSX-WINDOW上で効率よく実行できます。初めてSX-WINDOW用のプログラムに挑戦する人にも、簡単に基本機能の理解ができる33種のサンプルプログラム付き。また各マネージャ解説と関数リファレンスの詳細なマニュアルも装備しています。

※本ソフトのご使用に際しては、メインメモリ4MB以上、SX-WINDOW ver2.0以上、C compiler PRO-68K ver 2.1が必要です。





## キット構成

### ■開発ツール

**●SXデバッガ**

SX-WINDOW上で複数のプログラムを同時にデバッグすることができるソースコードデバッガ。

**●リソースエディタ**

SX-WINDOW上のリソースをリソースタイプごとの編集ウィンドウでビジュアルに作成・編集が可能。

**●リソースリンカ**

Cコンパイラやアセンブラで作成したリソースデータファイル(オブジェクトファイル)をリンクしてリソースファイルを作成。

**●サンプルメイク**

サンプルプログラムのコンパイル作業をSX-WINDOW上から、XCver2.1のMAKE.Xを呼び出して、自動実行する簡易メイクユーティリティ。

### ■サンプルプログラム

**●基礎編(23種)**

各マネージャの基本的な機能のみを用いた基本動作の理解。

**●応用編(4種)**

基礎編での基本機能を応用した簡単なアプリケーションの作成。

**●実用編(6種)**

基礎/応用編での機能を駆使した、実用的なアプリケーションの作成。

### ■その他のファイル

**●インクルードファイル**

Cコンパイラとアセンブラ用の関数定義、データ定義ファイル。

**●ライブラリファイル**

Cコンパイラ用の関数ライブラリ。

### マニュアル

●ユーザーズマニュアル●プログラマーズマニュアル●ファンクションリファレンス●ライブラリリファレンス



## X68000
PERSONAL WORKSTATION・XVI
## Compact

**本体＋キーボード＋マウス**
2HD3.5インチFDDタイプ
CZ-674C-H(グレー) 標準価格298,000円(税別)

**14型カラーディスプレイ(ドットピッチ0.28mm)**
CZ-608D-H(グレー) 標準価格94,800円(税別)



●5.25インチ増設用フロッピーディスクドライブ CZ-6FD5 標準価格99,800円・税別〔接続ケーブル同梱〕

●ディスプレイテレビ/CZ-6TU用RGBケーブル CZ-6CR1 標準価格4,500円・税別

●ディスプレイテレビ/CZ-6TU用テレビコントロールケーブル CZ-6CT1 標準価格5,500円・税別

●SCSI変換ケーブル CZ-6CS1 標準価格12,000円・税別

資料請求券 CZソフト 6h／X 1係

開いてくださいウィンドウ、触れてくださいインテリジェンス。

# さらに拡がる、SXワールド。

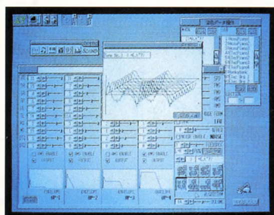
●多彩なサウンドクリエイトを実現するFM音源サウンドエディタ。

## SOUND SX-68K

NEW

CZ-275MWD 標準価格15,800円(税別)

他のミュージックソフトで演奏中の音色を、簡単に作成・変更ができるマルチタスク機能、またエディット、イメージ、ウェーブの3つの編集/確認モードを装備。作成中の音色も50曲の自動演奏でリアルタイムに確認、編集できます。まさにミキサー感覚で音創りが楽しめるツールです。 (2MB、ver1.1)



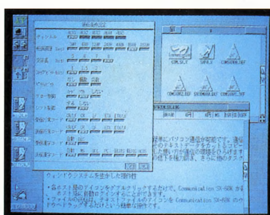
●マルチタスク機能をはじめ、通信環境がさらに充実。

## Communication SX-68K

NEW

CZ-272CWD 標準価格19,800円(税別)

通信環境をさらに高めたウィンドウ対応の通信ソフトです。マルチタスク機能により他のアプリケーションソフトを実行中でも簡単に通信が可能。また、ホスト局をクリックするだけの自動ログイン機能、初心者にも簡単なプログラム機能、最新モデム(20種類)もフルサポートしています。 (2MB、ver1.1)



●アウトラインフォント対応、ひらかれたウィンドウ環境。

## SX-WINDOW ver2.0

CZ-287SS 標準価格12,800円(税別)

フォントマネージャを装備してアウトラインフォントに対応、画面スクロール機能によるワイドデスクトップをはじめ便利機能を満載。 (2MB)

※SX-WINDOW ver1.0およびSX-WINDOW ver1.1をお持ちのかたには有償バージョンアップを行います。

●SX-WINDOWがより多彩に使えるツール集

## SX-WINDOW用ツール集

NEW

CZ-290TWD 12月発売予定

SX-WINDOWをさらに便利に、楽しく使うためのツール集です。「キーノーツ」、「スクリーンセーバ」、「スクラップブック」をはじめ、電子システム手帳リンク関連の「電子手帳通信ツール」、「アドレス帳」、「スケジューラ」など12個のツールが収まっています。 (2MB、ver2.0)

●簡単操作のウィンドウ対応グラフィックツール。

## Easypaint SX-68K

CZ-263GWD 標準価格12,800円(税別)

マウスによる簡単操作、65,536色中16色表示の多彩な表現。同時に複数のウィンドウを開いて編集でき、各ウィンドウ間でのデータ交換もできます。 (2MB、ver1.1)

●「SX-WINDOW開発キット」のサポートツール

## 開発キット用ツール集

NEW

CZ-289TWD 平成5年1月発売予定

SX-WINDOW開発キットをさらに使いやすくするためのツールです。SXコールのリファレンスを収めた「インサイドSX」、コードリソース作成のためのコンバータ「ハイパーCV」、アプリケーションのインストールが簡単に行える「インストーラ」をはじめ12種のツールが用意されています。 (2MB、ver2.0)

※(2MB、ver1.1)の表示は、メインメモリ2MB以上、SX-WINDOW ver1.1以上が必要であることを示します。

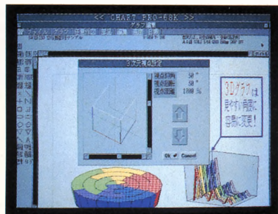
## 充実のPROシリーズ

●ビジネスグラフチャート

### CHART PRO-68K

CZ-267BSD 標準価格38,000円(税別)

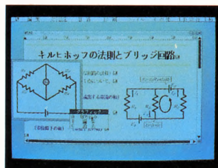
各種データベースで作成したデータをもとに、多彩なグラフが作成できます。3次元表示やグラフの複合機能も装備。データはMultiword,Press Conductor PRO-68Kに取り込むこともできます。



●グラフィック機能搭載の本格派ワープロ

### Multiword ver1.1

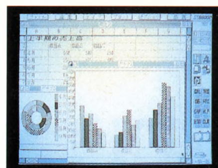
CZ-225BSD 標準価格32,000円(税別)



●簡単操作の統合型表計算ソフト

### BUSINESS PRO-68K Popular

CZ-286BSD 標準価格28,000円(税別)



●各種ドライバ、ライブラリを追加

### COMPILER PRO-68K ver2.1

CZ-285LSD 標準価格44,800円(税別)

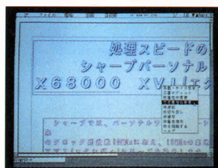


※有償バージョンアップ対応中。

●各種エディタ装備のレイアウトソフト

### PressConductor PRO-68K

CZ-266BSD 標準価格28,000円(税別)



※以上のPROシリーズのソフトの動作にはメインメモリ2MB必要です。

※発売予定のソフトの画面写真は実物とは異なる場合があります。

MONTHLY SPECIAL

# いまいちばん熱い 10万円台DOS/Vパソコン

もはや高嶺の花ではなくなった486のDOS/Vマシン。現在では各メーカーの主力機は10万円台で攻防を繰り広げているのだ。今月の特集は、手軽に購入できて高機能な10万円台486マシンを徹底的にレビュー。




BEST BUY 93

## カラー液晶搭載ファーストクラスのノートパソコンは?
## パーソナルに使えるベストな低価格のページプリンタは?
## 強豪ひしめくWindowsワープロのベスト1は?

特別企画

## いよいよ発表！読者と月刊PCで選ぶ最優秀パソコン
# 第1回PC OF THE YEAR

月刊PCは、"コストパフォーマンスに優れ、次の時代へつなぐ先進性を持った製品"を選びます。

対象ジャンル

[本体] デスクトップ/ノートブック
[周辺機器] プリンタ/ハードディスク/光磁気/増設メモリ/ディスプレイ/モデム
[ソフトウェア] ワープロ/表計算/データベース/統合/通信/言語/グラフィックス/エディタ/ユーティリティ




PC SYSTEM UP!

## TOWNSⅡをMPC仕様にし、日本でいちばん先進のパソコンに仕立てる

好評連載

- ●MORE REVIEWS～ツール、ユーティリティ、ボード、β版ソフトなどをレビュー
- ●パソコンAV塾～マルチメディア時代のパソコンシステム構築術
- ●HARD TUNEUP!～あなたの386マシンはまだまだ速くなる
- ●SOFT TUNEUP!～一太郎Ver.4をもっと快適に使いたい
- ●COLUMNS～水玉螢之丞/伊藤ガビン/富田倫生/竹山正寿/大谷和利/中尊寺ゆつこ

特別付録
①月刊PC特製パソコンダイアリー
②3.5インチFDラベル

SOFT BANK

## パーソナルコンピュータ総合情報誌
# 月刊PC

1月号/12月18日発売/特別定価650円(税込)/毎月18日発売

brother

パソコンソフト自販機 TAKERU

# ILLUSION CITY 幻影都市

〜イリュージョンシティ〜

©1991 MICRO CABIN CORP.

——科学が創る影なる都。——

ILLUSTRATION by YUKIOKITTA/CHARACTER DESIGN by 百瀬

禍々しき気に満ちた近未来都市、香港。狂気と悪しき欲望とが渦巻くこの都市を、いま一人の男が駆け抜ける。失われた己の過去を求めて、迫り来る危険に自ら身を投じる男、対魔掃討者"天人"は、人民警察の対魔特別攻撃班に属する女、"美紅"と共に、その実体さえ知れぬ巨大な悪に対し、渾身の気を込めて愛用の銃を放つ。果てしなく続く戦いの日々は、いつしか眠ることさえ忘れさせてしまった……

2月下旬発売予定







「サイバーパンク!!超伝奇RPG「幻影都市」

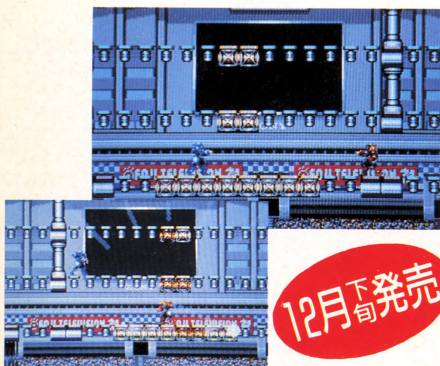
■8等身キャラクタ採用 ■キャラクタ演出革命!!
■ジョイパット&マウスオペレーション可能
■VRシステムVer.2.5搭載 ■MIDI対応

TAKERU価格 ¥6,800(税込)
■対応機種/X68000版
■制作/TAKERUソフト

©マイクロキャビン

## ストライクレンジ

サイドビュー、縦横スクロールのロボット対戦シューティングアクション、何層もの床で構成された近未来スタジアムで、今、最も危険なスポーツが始まった!ロボットの種類は8体、2人対戦モード付き、迫熱興奮のバトルに挑戦だ!



12月下旬発売

TAKERU価格 ¥4,800(税込)
■対応機種/X68000版
■制作/ギミックハウス

## 機甲装神 ヴァルカイザー

近未来、エネルギーを増幅する人工外皮「バイオローダー」の研究に伸びる黒い魔手。岬博士と妹留奈に襲いかかる者達の正体は…?美少女とメカとアニメーションといえば、ご存知「サイレンス」!初めてのX68000移植版がついに登場!もちろんフルアニメーションが、ガンガン入ってます!!



1月20日発売

TAKERU価格 ¥4,800(税込)
■対応機種/X68000版
■制作/サイレンス

## パチンコワールド

X68000オリジナルパチンコシミュレーション。音楽、グラフィックともに文句なしの出来の良さ!台の数は70以上。ダイヤル固定の為の硬貨アイテムを手にすればこわいものなし。怪人にさらわれた恋人を救い出すため、40台打ち止めに挑戦だ!!





好評発売中

TAKERU価格 ¥4,800(税込)
■対応機種/X68000版
■制作/ISC

ブラザー工業株式会社 〒467 名古屋市瑞穂区苗代町2番1号

TAKERU事務局 (052)824-2493

東京営業所(03)5203-7133
大阪営業所(06)252-4234

通信販売

通信販売をご希望の方は、ソフト名・機種名・住所・氏名・電話番号を明記の上TAKERU事務局まで現金書留でお申し込み下さい
代金引換は一度、現金書留で申し込んで頂いた方に案内させて頂きます

アフターサービス万全
　全商品保証付。専門の担当がお客様の立場で対応します。
初期不良、輸送トラブルetc.
万が一初期不良、輸送トラブルが発生しました際には、即交換させていただきます。

★頭金なし!!
★即日発送!!



# 秋葉原でおなじみのP&Aがズバリ超特価セールでご奉仕!!

- お近くの方は、お立寄下さい。専門係員が説明いたします。
- 本体単品でも受付します。詳しくは、お電話にてお問合せ下さい。
- ビジネスソフト定価の15%引きOK!! TEL下さい。
- 現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合せ下さい。

# 全国通販

## X68000用ソフトコーナー (送料1ヶ～5ヶまで¥500・消費税別)

| 商品 | 定価 | 特価 |
|---|---|---|
| ◆Z's STAFF PRO68 Ver3.0(ツアイト) | ¥58,000 | ¥37,500 |
| ◆Z's TRIPHONY デジタルクラフト(ツアイト) | ¥39,800 | ¥27,000 |
| ◆テラッツォ(ハミングバード) | ¥19,400 | ¥13,600 |
| ◆マジックパレット(ミュージカルプラン) | ¥19,800 | ¥14,200 |
| ◆たーみのる2(SPS) | ¥17,800 | ¥13,000 |
| ◆Mu-1 Super | ¥39,800 | ¥28,500 |
| ◆CMA68K(シティソフト) | ¥29,800 | ¥21,800 |
| ◆サイクロンEXPRESSα68 | ¥98,000 | ¥69,000 |
| ◆C-TRACE68 Ver3.0(キャスト) | ¥98,000 | ¥68,500 |
| ◆G68K Ver.2 PRO | ¥22,000 | ¥17,300 |
| ◆C&Professional Pack V3.2(マイクロウェアジャパン) | ¥80,000 | ¥57,800 |
| ◆ウエットペイント1～3(ウエーブトレイン)〔各〕 | ¥15,000 | ¥11,500 |
| ◆マチエール(サンワード) | ¥39,800 | ¥28,800 |
| ◆Windex PRO68(JEL) | ¥28,000 | ¥20,500 |
| ◆CZ-213MSD MUSIC PRO68K | ¥18,800 | ¥13,200 |
| ◆CZ-214MSD SOUND PRO68K | ¥15,800 | ¥11,300 |
| ◆CZ-215MSD Sampling PRO68K | ¥17,800 | ¥12,500 |
| ◆CZ-220BSD DATA PRO68K | ¥58,000 | ¥40,000 |
| ◆CZ-224LSD The 福袋 Ver2.0 | ¥9,900 | ¥7,400 |
| ◆CZ-225BSD Multiword Ver1.1 | ¥32,000 | ¥23,000 |
| ◆CZ-243BSD CYBERNOTE PRO68K | ¥19,800 | ¥15,000 |
| ◆CZ-247MSD MUSIC PRO68K〔MIDI〕 | ¥28,800 | ¥20,500 |
| ◆CZ-249GSD CANVAS PRO68K | ¥29,800 | ¥22,000 |
| ◆CZ-251BSD Hyper word | ¥39,800 | ¥29,400 |
| ◆CZ-253BSD CARD PRO68K Ver2.0 | ¥29,800 | ¥22,700 |
| ◆CZ-257CSD Communication PRO68K Ver2 | ¥19,800 | ¥15,300 |
| ◆CZ-258BSD Teleportion PRO68K | ¥22,800 | ¥16,900 |
| ◆CZ-261MSD MUSIC studio PRO68K Ver2.0 | ¥28,800 | ¥21,200 |
| ◆CZ-263GWD Easypaint SX-68K | ¥12,800 | ¥9,800 |
| ◆CZ-265HSD New PrintShop Ver2.0 | ¥20,000 | ¥15,400 |
| ◆CZ-266BSD PressConductor PRO68K | ¥28,800 | ¥22,000 |
| ◆CZ-267BSD CHART PRO68K | ¥38,000 | ¥29,800 |
| ◆CZ-284SSD OS-9/X68000 Ver2.4 | ¥35,800 | ¥25,600 |
| ◆CZ-285LSD C-Compiler PRO68K Ver2.1 | ¥44,800 | ¥32,500 |
| ◆CZ-286BSD BUSINESS PRO68K Popular | ¥28,000 | ¥20,500 |
| ◆CZ-287SS SX-WINDOW Ver2.0 | ¥12,800 | ¥9,800 |

★ゲームソフト25%OFF!!(一部ソフト除く)

## 周辺機器コーナー (送料¥500・消費税別)

| 商品 | 定価 | 特価 |
|---|---|---|
| ①CZ-8NS1 | ¥188,000 | ¥133,000 |
| ②CZ-6VT1 | ¥69,800 | ¥49,500 |
| ③CZ-6TU | ¥33,100 | ¥23,900 |
| ④BF-68PRO | ¥19,800 | ¥14,400 |
| ⑤CZ-8NM3 | ¥9,800 | ¥7,200 |
| ⑥CZ-8NT1 | ¥13,800 | ¥10,000 |
| ⑦CZ-6BE2A | ¥59,800 | ¥42,800 |
| ⑧CZ-6BE2B | ¥54,800 | ¥39,300 |
| ⑨CZ-6BE2D | ¥54,800 | ¥39,300 |
| ⑩CZ-6BF1 | ¥49,800 | ¥35,800 |
| ⑪CZ-6BP1 | ¥79,800 | ¥57,000 |
| ⑫CZ-6BM1 | ¥26,800 | ¥19,300 |
| ⑬CZ-6EB1 | ¥88,000 | ¥63,000 |
| ⑭AN-S100 | ¥36,600 | ¥26,300 |
| ⑮CZ-6SD1 | ¥44,800 | ¥32,500 |
| ⑯CZ-6BN1 | ¥29,800 | ¥21,500 |
| ⑰CZ-6BV1 | ¥21,000 | ¥15,200 |
| ⑱CZ-6BC1 | ¥79,800 | ¥57,000 |
| ⑲CZ-6BG1 | ¥59,800 | ¥43,000 |
| ⑳CZ-6BU1 | ¥39,800 | ¥28,500 |
| ㉑CZ-6PV1 | ¥198,000 | ¥142,000 |
| ㉒CZ-6BS1 | ¥29,800 | ¥21,500 |
| ㉓CZ-8NJ2 | ¥23,800 | ¥17,500 |
| ㉔CZ-6BL2 | ¥298,000 | ¥214,000 |
| ㉕JX-100S | ¥89,800 | ¥44,000 |
| ㉖JX-220X | ¥168,000 | ¥121,000 |
| ㉗IO-735XB | ¥248,000 | ¥152,000 |
| ㉘LC-10C1H | ¥598,000 | ¥459,000 |
| ㉙CZ-6CS1(674C用) | ¥12,000 | ¥8,900 |
| ㉚CZ-6CR1(RGBケーブル) | ¥4,500 | ¥3,600 |
| ㉛CZ-6CT1(テレビ・コントロール) | ¥5,500 | ¥4,400 |
| ㉜CZ-6BP2 | ¥45,800 | ¥33,300 |

## 中古・高価現金買取り 下取りOK!!

■まずはお電話下さい。
下取り専用 買取り電話 ▶ ☎03-3651-1884 FAX. 03-3651-0141
■下取り・買取りで、お急ぎの方は、直接当社に来店、または宅急便にてお送り下さい。

買取り価格…完動品・箱/マニュアル/付属品付の価格です。

- 下取りの場合……価格は常に変動していますので査定額をお電話で確認して下さい。(差額は、P&A超低金利クレジットをご利用下さい。)
- 買取りの場合……現品が着き次第、2日以内に買取り金額を連絡し、振込み、又は書留でお送り致します。
- 近郊の方は、P&A本店まで、直接お持ち下さい。即金にて、¥1,000,000までお支払い致します。

- 最新の在庫情報・価格はお電話にてお問い合せください。
- 買い取りのみ、または、中古品どうしの交換も致します。詳しくは電話にて、お問い合せ下さい。
- 価格は変動する場合もございますので、ご注文の際には必ず在庫をご確認下さい。
- 本商品の掲載の価格については、消費税は、含まれておりません。
- 現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合せ下さい。

## 《便利な超低金利クレジットをご利用下さい》

- 月々¥1,000円からOK!!
- ボーナス払いOK(夏冬10回までOK)
- 支払い回数 1回～84回
- お支払いは、8ヶ月先からでもOK!!

- 定休日/毎週水曜日

## P&A特選＝今月の中古特選品

| | 限定 | 限定 |
|---|---|---|
| ●CZ-601C ●CZ-611D-TN ¥120,000 | ●CZ-634C-TN ●CZ-606D-TN ¥198,000 | ●CZ-644C-TN ●CZ-604D-TN ¥298,000 |

## 買取り価格

| 機種 | 価格 | 機種 | 価格 |
|---|---|---|---|
| ●CZ-634C | ¥150,000 | ●CZ-602C | ¥68,000 |
| ●CZ-644C | ¥200,000 | ●CZ-612C | ¥78,000 |
| ●CZ-604C | ¥80,000 | ●CZ-652C | ¥48,000 |
| ●CZ-623C | ¥110,000 | ●CZ-662C | ¥68,000 |
| ●CZ-603C | ¥78,000 | ●CZ-611C | ¥58,000 |
| ●CZ-613C | ¥90,000 | ●CZ-601C | ¥45,000 |
| ●CZ-653C | ¥68,000 | ●CZ-674C | ¥150,000 |
| ●CZ-663C | ¥75,000 | | |

## 下取り交換差額表

| 下取り＼新品 | CZ-634C モニターセット | CZ-644C モニターセット | モデル UX20セット | モデル CX20セット | 9801FA2 |
|---|---|---|---|---|---|
| CZ-623C モニターセット | 150,000 | 270,000 | 70,000 | 160,000 | 140,000 |
| CZ-613C モニターセット | 190,000 | 290,000 | 100,000 | 190,000 | 170,000 |
| CZ-652C モニターセット | 230,000 | 340,000 | 150,000 | 240,000 | 190,000 |
| CZ-604C モニターセット | 180,000 | 290,000 | 100,000 | 190,000 | 150,000 |
| CZ-600C モニターセット | 230,000 | 340,000 | 150,000 | 240,000 | 200,000 |

## 通信販売お申し込みのご案内

〔現金一括でお申し込みの方〕
- 商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで、現金書留でお送りください。(プリンター・フロッピーの場合、本体使用機種名を明記のこと)

〔銀行振込でお申し込みの方〕
- 銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様のご住所・お名前・商品名等をお知らせください。(電信扱いでお振込み下さい。)

〔振込先〕さくら銀行 新小岩支店
当座預金 2408626 ㈱ピー・アンド・エー

〔クレジットでお申し込みの方〕
- 電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。
- 現金特別価格でクレジットが利用できます。残金のみに金利がかかります。
- 1回～84回払いまで出来ます。但し、1回のお支払い額は¥1000円以上。

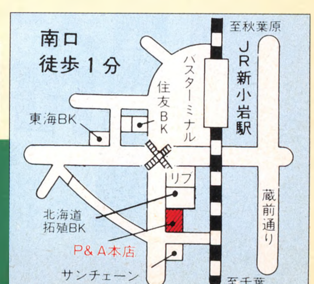
**超低金利クレジット率**

| 回数 | 3 | 6 | 10 | 12 | 15 | 24 | 36 | 48 | 60 | 72 |
|---|---|---|---|---|---|---|---|---|---|---|
| 手数料 | 3.0 | 4.0 | 5.5 | 5.5 | 8.5 | 11.5 | 16.0 | 21.0 | 27.0 | 33.0 |

注目!!平成5年3月末一括払い手数料(金利)無料(平成5年1月末/2月末/3月末のいずれかをご指定下さい。)

マイコン専門ショップ **P&A**
株式会社ピー・アンド・エー
〒124 東京都葛飾区新小岩2丁目1番地19号
☎03-3651-0148(代) FAX. 03-3651-0141

営業時間
平日:AM10:00～PM7:00
日祭:AM10:00～PM6:00




- 価格は流通事情により変動致しますので、銀行振込・書留等の送付前に、あらかじめお電話にてご確認下さい。

臨時増刊　12月17日発売

予価 380円

ーコンまるかじり!

# The スーパーファミコン

## 1冊まるごとRPGだ!

スーパーファミコンRPGファン待望

話題の新作RPG、RPG読み物企画満載

## 「ファイナルファンタジーV」総集編




新作ガイド
**エルファリア**
**BURAI 八玉の勇士伝説**
**ノイギーア** ほか

ドリィV/レナス ほか

**モンスターガイド**

集

スーパ
これからどのRPGを買おうかと悩んでいる


SOFT BANK
ソフトバンク出版事業部

親切と安さのTSUKUMO
1992から1993に向けて

# ファイナルダッシュセール

初売は3日より、特価品を揃えてお待ちしております。掲載商品2万円以上 送料無料(一部地域を除く)

**秋葉原電気まつり**
4年11月20日金～5年1月7日木
賞金総額6,000万円
★1等 現金10万円 ★2等 現金5万円
★3等 現金1万円 ★4等 現金5千円
※店頭にて5,000円お買い上げ毎に1枚、抽選券をプレゼント!

## シャープX68000の事なら何でも揃うツクモにおまかせ!

### X68000いろいろ組み合わせ提案いたします

●X68000の未来を象徴するハイコンパクトなボディ(体積比44%)●成熟するウィンドウ環境、使いやすさと高機能を追求したSX-WINDOW Ver2.0搭載●2HD 3.5インチフロッピーディスクドライブ2基搭載●カラー液晶ディスプレイ接続可能●X68000XVIの高性能を継承●VGAモードサポート(SX-WINDOWのみ対応)

X68000 Compact XVI

秋葉原を歩き回る必要はありません。情報が沢山。分らない事は何でもお尋ね下さい。目に優しい10.4型カラー液晶ディスプレイ(LC-10CI)も取り扱い中!詳しくはお問い合わせ下さい。
システムのご相談は
☎03(3253)1899
までどうぞ!!

●5インチソフトも使える欲張りセット
CZ-674C-H(X68000 Compact本体)……定価¥298,000
CZ-608D-H(0.28mmピッチカラーディスプレイ)定価¥94,800
5インチドライブ(2ドライブ)……サービス
ツクモ特価¥336,000

●ハードディスクで便利に使えるセット
CZ-674C-H(X68000 Compact本体)……定価¥298,000
CZ-608D-H(0.28mmピッチカラーディスプレイ)定価¥94,800
100MBハードディスク……サービス
ツクモ特価¥348,000

### ツクモオリジナル
目のつけどころがツクモでしょ。

●X68000 シリーズ専用3.5インチフロッピーディスクドライブ
**TS-3XRシリーズ**
●3.5インチ2DD/2HD/2HC/1.44MBフォーマット対応(いろいろなフォーマットのメディア読み書きが出来ます。
●ユーティリティソフト付属(デバイスドライバー/フォーマッター)
TS-3XR1(1ドライブ) 定価¥44,800 ツクモ特価¥35,800
TS-3XR2(2ドライブ) 定価¥57,800 ツクモ特価¥46,800

●X68000 Compact XVIシリーズ用5インチフロッピーディスクドライブ
**TS-5XRシリーズ**
●5インチ2HD/2DDフォーマット対応●ドライブ番号切り換えスイッチ付
TS-5XR1(1ドライブ) 定価¥53,800 ツクモ特価¥42,800
TS-5XR2(2ドライブ) 定価¥72,800 ツクモ特価¥57,800
※写真はTS-3XR1です。

### 大好評 Matierセット

**その1** 慣れるとマウスがいらない
WACOM製
●SD-510C タブレット……¥98,000
●TJ-410A-2 拡張ケーブル……¥6,000
●SP-200A スタイラスペン……¥10,000
サンワード
●Matier(マチエール)……¥39,800
合計定価¥153,800
ツクモ特価¥128,000

**その2** ハイクオリティなのにこんなに安い
ヒューレットパッカード HP Desk Jet 505J
インクジェットプリンタ¥99,800
カラーキット……¥12,000
アーベル プリンタケーブル……¥4,800
サンワード Matier(マチエール)……¥39,800
合計定価¥156,400
ツクモ特価¥123,000

**その3** ベストセラーのBJプリンターがカラー対応に!!
キャノン BJ-15V……¥89,800
カットシートフィーダ……¥9,800
プリンタケーブル……¥4,800
カートリッジ……¥15,000
(Y、M、C、それぞれ1個ずつ)
サンワード Matier(マチエール)……¥39,800
合計定価¥159,200
ツクモ特価¥123,000

NEW
**耳よりな情報**—X68000XVI/Cnmpact XVIシリーズお持ちの方
2MB増設RAM(CZ-6BE2Bコンパチ)
既に、内蔵メモリーボードを搭載して4MBに増設されている方で、更に増設をお考えの方へお勧め商品
TS-6BE2B この冬特別限定生産 ツクモ特価¥34,800

### SCSIタイプハードディスク
VIP 100CX(100MB ダークグレー)ツクモ特価¥63,000
VIP 120CX(120MB ダークグレー)ツクモ特価¥73,000
LHD-FM200E(200MB) ツクモ特価¥98,000
LHD-FM240(240MB) ツクモ特価¥118,000
※SCSIボード(CZ-6BS1 定価¥29,800)は別売です。

ラストチャンス!!
SASIタイプハードディスク 限定
SHD-40J
ツクモ特価¥49,800

超低金利!冬・夏ボーナス二回払受付中!!詳しくは各店までお問い合わせ下さい。

### MIDI コンピュータミュージック特選セット

**特選Aセット**
●SC-55……¥69,000
●SX-68M-II……¥19,800
●Mu-1 SUPER……¥39,800
合計定価¥128,000
ツクモ特価¥99,000
(消費税別途¥2,970)
クレジット例(18回払・税込)
初回¥6,596+月々¥6,300×17回

**特選Bセット**
●CM-300……¥58,000
●SX-68M-II……¥19,800
●Mu-1 SUPER……¥39,800
合計定価¥117,600
ツクモ特価¥92,000
(消費税別途¥2,760)
クレジット例(10回払・税込)
初回¥10,919+月々¥10,000×9回

**特選Cセット**
●CM-500……¥115,000
●SX-68M-II……¥19,800
●Mu-1 SUPER……¥39,800
合計定価¥174,600
ツクモ特価¥141,000
(消費税別途¥4,230)
クレジット例(15回払・税込)
初回¥11,300+月々¥10,500×14回

**特選Dセット**
●SC-33 ¥49,800
●SX-68M-II ¥19,800
●Mu-1 SUPER ¥39,800
合計定価¥109,400
ツクモ特価¥88,000
(消費税別途¥2,640)
クレジット例(10回払・税込)
初回¥10,131+月々¥9,600×9回

### メモリーボード
■1MB増設RAMボード(CZ-600C専用) ツクモ特価¥19,500
■1MB増設RAMボード(ACE/PRO/PRO2シリーズ用) ツクモ特価¥17,000
■2MB増設RAMボード(拡張スロット用) ツクモ特価¥33,800
■4MB増設RAMボード(拡張スロット用) ツクモ特価¥59,800
※計測技研のメモリーボードも取り扱っております。価格についてはお尋ね下さい。

### SX-WINDOWワールド
●SX-WINDOW開発キット
CZ-288LWD 12月発売予定
●SX-WINDOW Ver.2.0
CZ-287SS ¥12.800
●Communication SX-68K
CZ-272CWD ¥19,800
●SOUND SX-68K
CZ-275MWD ¥19,800
●Easypaint SX-68K
CZ-263GWD ¥12.800

### 電子文具
タイムマネージメントを管理する便利ツール
●従来の電子システム手帳用ICカードがそのまま使えます●次から次へと忙しい方の為の強力な助っ人●大画面・大容量・手書き入力で操作効率向上!
シャープ 電子マネージメント手帳
PV-F1 定価¥128,000
ツクモ特価販売中!
更に、便利な名刺読み取り機「PV-BR1」(標準価格¥120,000)もお勧めです。

### パソコン通信
時代は9600ボーへ!!
■モデム 9600bps MNP5 & CCITT V.42bis ツクモ特価¥49,800～
■通信ソフト た～みのる2 ツクモ特価¥14,000

### 液晶ビジョン
あなたの部屋がミニシアター&迫力ゲームセンターに変身!
シャープ液晶ビジョンセット
XV-P1 定価¥220,000
今ならRGB信号→S端子変換ユニットプレゼント
ツクモ特価¥198,000

### X68000用MOディスク
ツクモはSONY MOディスクの正規代理店です。
これが今一番の人気者!
SONY 3.5インチ光磁気ディスクユニットセット
●RMO-S350(3.5"光磁気ディスクドライブ)¥235.000
●SCSIケーブル……¥6.900
●SCSIインターフェースボード……¥29.800
合計定価 ¥271.700
ツクモ特価販売中

通信販売のご注文は下記フリーダイヤルへ。
**全国どこからでも通話料無料**
受・注・専・用 フリーダイヤル 0120-377-999
通販センター 03-3251-9911 商品についてのお問い合わせは各店又は通販へ。

| | | |
|---|---|---|
| **クレジット払い** 月々¥3,000以上の均等払いも頭金なし、夏・冬ボーナス2回払いも受付中! | **各種リース払い** くわしくは各店にお問い合わせ下さい。ケースに合わせてご相談にのります! | **現金書留払い** 〒101-91 東京都千代田区神田郵便局私書箱135号 ツクモ通販センター Oh!X係 |
| **カード払い**(¥5,000以上) 通信販売での御利用カード、ツクモグローバルカード、VIPカード、セントラル、ジャックス※御本人様より電話で通販部へお申し込み下さい。 | **全国代金引き換え配達** お申し込みは☎03-3251-9911へお電話1本! 配達日の指定もできます。 | **銀行振込払い** 事前に☎でお届け先をご連絡下さい。三和銀行 秋葉原支店(普)1009939 ツクモデンキ |

秋葉原各店
営AM10:15～PM7:00
▲至お茶ノ水 昌平橋通り パソコン本店 牛丼屋 シントク電気 万世橋警察 三菱銀行 中央通り ラジオセンター ニューセンター店 AV/カメラ館 駐車場 JR 秋葉原駅 JR山手・京浜東北線 ◀至神田 至上野▶ ▼至浅草橋

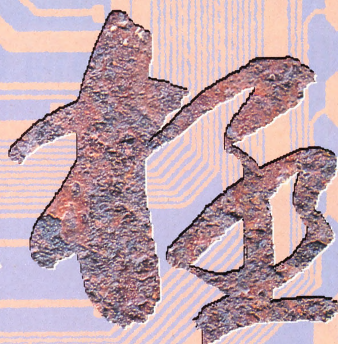
ツクモは「スーパーX PRO SHOP」です。
PRO STAFF **ツクモ**
九十九電機株
〒101-91 東京都千代田区神田郵便局私書箱135号

**ツクモパソコン本店2F**
☎03-3253-1899(直通)
パソコン本店代表 ☎03-3253-5599(担当/荒井)休毎週木曜
■ツクモニューセンター店 ☎03-3251-0987(担当 沢栄)休毎週木曜
■名古屋1号店 ☎052-263-1655(担当 山口)休毎週火曜
■名古屋2号店 ☎052-251-3399(担当 松原)休毎週水曜
■ツクモ札幌店 ☎011-241-2299(担当 田口)休毎週木曜
■DEPOツクモ2番店 ☎011-242-3199(担当 鈴木)休毎週木曜
※12月は無休で営業致します。

★商品のご注文は在庫確認の上お願いします ★表示価格には消費税は含まれておりません

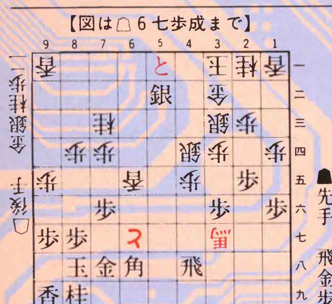
# 世界最強の将棋ソフト X68000に新登場!!

PC-98版・FM-TOWNS版と大好評の「極」がX68000で、その狂暴性を発揮。特徴は、もちろんその強さ、さらにフロッピー1枚で約500局もの棋譜が管理できます。また、高速で詰将棋を解かせたり、定跡の編集などにも最適です。
※「極」は狂暴なため、取扱いには充分ご注意を。



## 以下はX68000PROで解かせた「次の一手」です。

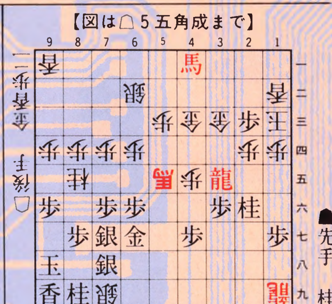
●難易度
●正解
●「極」解答時間／「極」思考レベル




●三段クラス
●先手1三角成
●1分53秒/ノーマル4
以下後手同香，先手1二飛




●二段クラス
●先手3三飛
●3分56秒/ノーマル5
以下後手2六歩，先手2三飛成




●初段クラス
●先手1五龍
●1分28秒/ノーマル4
以下後手同歩，先手3一馬
後手2三玉，先手3五桂

●例題は「終盤の定跡デラックス」(週刊将棋編・毎日コミュニケーションズ発行)より

### 攻撃的、だから面白い!

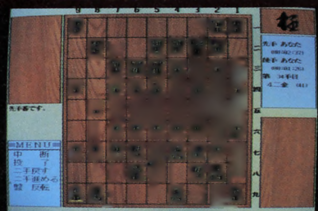
●「極」は打倒有段者を目標に開発を行ってきました。今までのコンピュータ将棋では守りを重視しがち　「負けなければよい」という印象は否めませんでした。しかし「極」に搭載された新開発の局面解析と正確な攻撃アルゴリズム・好手選択ルーチンは、攻めにも重きを置き、「勝たなければならない」ソフトに仕上がっています。そしてこれからのコンピュータ将棋の方針になるでしょうし、本来あるべきだった姿ともいえます。なにより攻守にわたるかけひき、一手を争う終盤戦と、将棋のたのしみがなければ始まらないではありませんか。「極」の最強モードは有段者と互角に戦うことができると自負しております。

### 強い、だから速い!

●従来の将棋ソフトは、最高レベルの思考時間の長さが問題でした。つまるところ、長すぎる!しかし、「極」はノーマルレベルにあってすでに従来の最高クラスの実力をお見せできます。そしてこれはとりもなおさず速いということ。従来並の実力をお求めの方には処理スピードの速さをもご紹介しておきましょう。(詰将棋に関しても、市販ソフト中最高速です)
●「極」の売りはその強さ。X68000版での最強モード・「レベル5」では、一局に約1日かかってしまいます。しかし他ソフトを凌駕するこの実力は必ずご納得いただけるものと考えています。

**X68000版新発売!!**
**PC-9801版発売中!!**
**FM-TOWNS版発売中!!**
**「極」定跡集各機種とも定価4,800円で好評発売中!!**





▲いままでのソフトの棋力に不満だったユーザーにも自信をもってお勧めできる極の攻撃力。

**定価12,800円**

シャープ電子システム手帳用ICカード

## 「篠沢教授の超脳力クイズ、天才秀才ただの人」

[品名／イベントクイズカード 形名／PA-3C48S] 標準価格¥8,500(税別)




### 特徴

- テレビでおなじみ学習院大学の篠沢秀夫教授とそのスタッフがあなたに挑戦!
- Y/N、3択、記述式と変化に富んだ問題を約3,000問用意。
- ゲームは1人用のチャレンジモードと、2人用対戦モードを登載。また対戦用は実力に応じてハンディを付けられ、俄然盛り上がります。
- 途中で電源を切っても再開できるコンティニューモードや、カードを抜いても再開できるパスワード機能を登載(チャレンジモードのみ)
- クイズの途中には、様々なイベントが用意されています。
- 最後の超難解なクイズ50問をクリアした方のうち先着50名をクイズ神社で名高い「久伊豆神社(埼玉県岩槻市)」にお名前を奉納させて頂きます。

テレビでおなじみの篠沢教授とそのスタッフが電子手帳用に考えた問題約3,000問のクイズゲームです。正解に応じてビル(3つ)を登って行き、クイズの途中には神経衰弱や競馬ゲームなど様々なイベントゲームを設けておりますので、クイズの得意な人から苦手な人まで、幅広くゲームを楽しんでいただけます。

LOG
TEL. 03-3837-2595 代
〒110 台東区台東2-4-3:4F

Oh!X **Graphic Gallery**

## DōGA CGAマガジン創刊

ここで使われているオブジェクト，マッピングデータはすべてCGAマガジンに収録されているもの。モーションブラーをかける"ような"ツールも入っている。

作品の発表の場とデータベースの構築を目的として，DōGA CGAシステム用のデータ集が創刊される。データ集といっても，システムなども入っており，簡単にアニメーションが作成できる。また，バージョンアップしたツールなども収められている。

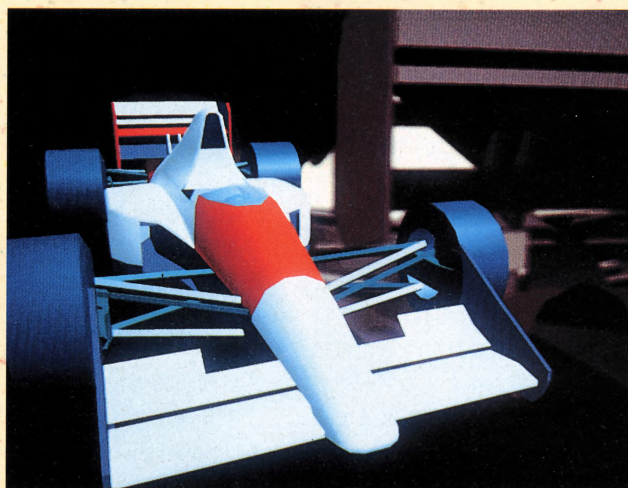
写真を見てのとおり，データのレベルはさすがというか，×××というか，正気の沙汰ではない。これだけでも一見の価値がある。

順調にいけば12月20日からタケルで入手できる。価格は1,000円だが，これはタケル使用料であって，基本的にCGAマガジン自体は無料，コピーフリーとし，ネットなどでも配布する予定。ただし無料とはいっても無償ではなく，カンパやオリジナルデータの提供，労働力の提供などは必要とのこと。

X68000用　3.5/5"2HD版
プロジェクトチームDōGA







細かくモデリングされたデータが集められている

# NICOGRAPH'92






























キン

ーションがある

## NICOGRAPH'92

今年もNICOGRAPH'92こと，コンピュータグラフィックス総合展が，池袋サンシャインシティ（フィルムショウなどは有楽町マリオン）で11月9日から13日まで開催された。

NICOGRAPHでの機器展示は，やはりワークステーション関連がメインとなるが，最近のパーソナルコンピュータでのCGの発達を受け，MacintoshやPC-9801，X68000などの姿も見かけられた。

しかし，いちばん派手に目についたのは，やはりシリコングラフィックスのブースであろう。手軽な製品から高度な製品までをカバーすべく，幅広い展開を見せていた。Indigo向けにはIllustlatorを展示したり，ペイントソフトを実費配布して，一般への普及を狙い，CrimsonではReality Engineで迫力のあるデモをリアルタイムで流していた。自社ロゴの入ったフリスビーを配っていたのもユニーク。

ダイキン工業のメインステージでは，「SOFT IMAGE」のデモンストレーションとして，寺尾響子さんが参加しているNHKの番組「DREAM」のキャラクター“音符くん”が登場。データグローブで制御されて，画面の中を飛び回っていた。また，アミューズメントへの応用例として「バーチャレーシング」が置かれており，長い列が見受けられた。

アミューズメント関連ではほかにナムコも「Sim Drive」を出品。ユーノスロードスターの姿に人が引き寄せられて人だかりができていたが，試乗する人はわりと少なかったようだ（照れがあるためか）。

そして，X68000はシャープのCAD事業部，メディックスの2つのブースでがんばっていた。前者は「μイメージシンセサイザ」のシステムとして展示されていたが，今回は以前紹介した標準システムのほかに，X68000Compactによるハイビジョン出力システムも参考出品されていた。後者は当然「MIRAGE System Model Stuff」を走らせての展示である。

お馴染みのバーチャルリアリティもバラエティ豊かな関連機器が展示されていたが，まだまだ発達段階。しかし，こうした各分野へのCGの応用は近い将来に花開き，フィードバックをもたらしながら，また別のかたちで実を結ぶものと思われるので，決して無駄なことではないだろう。











# Inter BEE'92







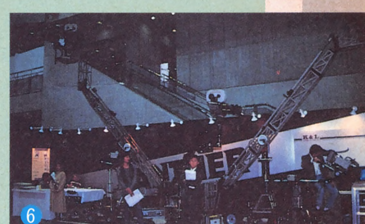















❶Video Toasterを巧みに操るKIKI嬢
❷こんなところにPerformar600が
❸❹Macintosh用とIBM PC用があるFast社のビデオ編集ボード「Video Machine」
❺こちらもMacintoshによるビデオ編集システム
❻国際放送機器展だからこういうのもある
❼これはVRではなく耐振動ハンディカメラクレーン
❽滝に打たれながら音を流し続ける全天候型スピーカー

## Inter BEE'92

「Inter BEE'92」とは聞き慣れない名前かもしれないが，「International Broadcast Equipment Exhibition」，つまり国際放送機器展の略称である。

放送局の技術者や番組制作会社など，放送技術に関心をもつ層を対象としているだけあって，一般の人々には馴染みがなくてもしかたないであろう。

しかし，最近はパーソナルコンピュータ用のビデオ編集システムも多く発売されており，なかにはブロードキャストクオリティ（放送局レベル）の機能，入出力装置を備えたものもある。

11月11日から13日まで幕張メッセで開催された「Inter BEE'92」でも，そのあたりの情勢が反映されているだろうということで足を運んでみた。

個人的に興味があったのは，「Video Toaster」の最新バージョン，および日本語版である。「Video Toaster」はAMIGAの内蔵ボードとして作られたビデオ編集支援システムで，AMIGAのマシンとしての寿命を数年延ばした，といっても過言ではない。X68000にもこういう周辺機器が出てきてほしいものだ。

ブースでは「Video Toaster」のスペシャリストであるKIKI嬢が，ひっきりなしにデモンストレーションを行っていた。ほかのブースを回っている間も声はずっと聞こえていたので，ほとんど休みなしの状態で操作していたようだ。休んでいるときは休んでいるときで，自分のブロマイドにサインをして来場者に配っていた。

また，イ・アイ・イが展示していたFast社のビデオ編集ボード「Video Machine」は，「Video Toaster」と同様の機能で，こちらはMacintosh用とIBM PC用の２つがある。「Video Toaster」に比べると値段はやや高めということになるが，本体の普及率や今後の展開を考えると，どちらに軍配が上がるかは微妙なところだろう。

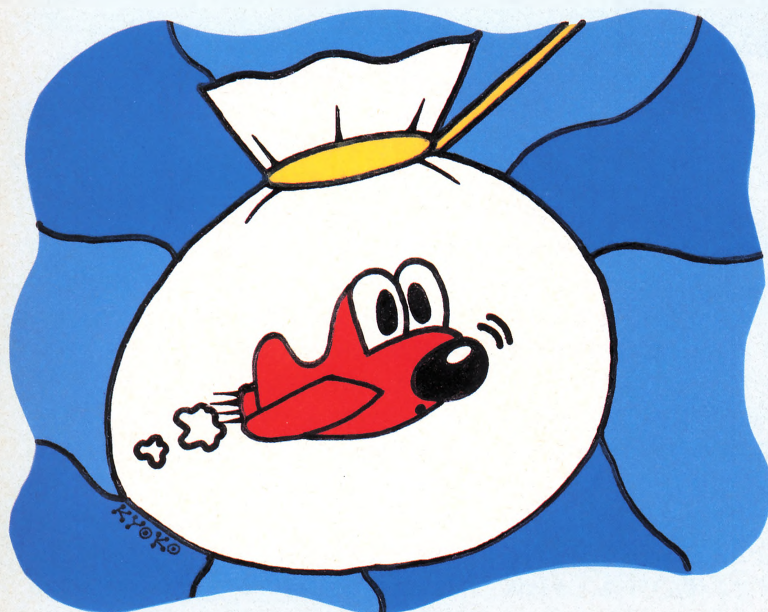
キヤノンの「VideoF/X」も含め，パーソナルコンピュータでの制御を前提としたビデオ編集システムは数十万円から百数十万円という値段である。パソコンの周辺機器として考えると少し値が張るのだが，同機能の放送用機器と比べるとかなり安価になる。会場でもそういう点でずいぶんと関心をもたれていたようである。　(R.A.)

［第20回］
## ジェットキンギョ

寺尾　響子

# 響子inCGわ〜るど

「きんぎょ〜え，きんぎょ」
　高らかな声が響いたのは，お正月も3日目の朝でした。ここは超高層マンションの214階。こんなところに金魚売りが。しかも冬に。なんで？
「え〜，きんぎょ〜え，きんぎょ」
　声はドアの外でいっそう冴えわたり，早く開けろといわんばかりです。静かな朝をじゃまされて少し腹が立ちましたが，放っておくとますます大きくなりそうなので応対することにしました。
　ドアを開けると，白髪頭に手ぬぐいをきりりと巻いた小柄な老人が，着物姿で立っていました。そしてそのわきには，唐草模様の風呂敷で覆われた四角いものが。
　老人はにっこりと笑って深々と一礼し，
「あけましておめでとうございます」
といいました。
　昔，東京の下町では新年のお祝いに獅子舞を呼び，ご祝儀を渡して玄関先で舞ってもらったそうだけど，その類かな。
「さてさて，ご覧あれ」
　老人が風呂敷をはらりとはずすと，ガラスケースが現れました。そのなかに朱色や銀色の生き物が威勢よく飛び回っています。
「ちょっとめずらしい生き物でして，ジェットキンギョといいます」
「じぇっとき……，んぎょ？」
「ええ，人工蛋白質が主成分の生き物でしてね，お尻のところがジェット噴射口になってます。それで飛ぶんですな。まあ，一種のサイボーグってえところですかね」
　耳を澄ますと，ジェット機独特の爆音が小さく聞こえました。朱色の1匹がこちらをじっと見つめています。どうやら知能もそこそこにあるようです。
　目を細めて老人がいいました。
「こいつらなかなか賢くて，イルカのように芸をしますんで」
と，胸元からうす紫色の小冊子を取り出して広げました。表紙にパープルという文字が見えます。
「ニイタカヤマノボレ！」
「これで次の指示を待つよう，全部のキンギョにセットされました」
「トトト……」
　魚という意味の幼児語かと思いましたが，違いました。とたんに鳥のように群れをなして，同じ方向に飛び始めたのです。
　このほかにも老人はいくつか号令をかけましたが，そのたびにキンギョたちはいろいろな行動をとるのでした。
　朱色と銀色が揃って動くさまはモザイクのようでとてもきれいだったので，なるほど正月にはふさわしいと思いました。
「さて，これが最後の号令です」
「ヒビコレケッセン！」
　キンギョたちは無邪気な顔で互いに体当たり攻撃を始めました。しばらくすると，1つひとつ落






ちていき，動かなくなっていきます。
　ヒビコレケッセン……ヒビコレケッセン……，日々是決戦。
　その言葉は，いま通っている某マンモス予備校のスローガンです。
　記憶の泡がはじけて飛びました。
　昨年，私は8つの大学入試を目前にした受験生でした。ひとつ目の入試を受けにいったときのことです。寒さに背を丸めて，会場に向かう受験生の群れに交じると，どっと疲れが押し寄せてきました。イヤダ。くるりと向きを変え，ひとり流れに逆らって歩きました。
　テキゼントウボウ。
　そして今年，浪人1年目のお正月を迎えたのでした。

＊　＊　＊

　顔に冷たいものが当たりました。1匹のキンギョがケースから飛び出したのです。
「おや，どうやらこいつはあなたのことが気に入ったようだ。よし！　新年のお祝いに差し上げましょう」
　老人はビニール袋にキンギョを入れて渡してくれました。そして，てきぱきとケースをしまうと足早に去っていきました。
　大急ぎで老人を追いかけましたが，もう影すら見えませんでした。ひとつ聞きたいことがあったのに……。
　コノキンギョ，ドウヤッテソダテレバイイノ？

# SOFTWARE INFORMATION

10月末から11月にかけて，たくさんの，しかも粒の揃ったゲームが発売された。そのあおりを受けたせいか，年末の新作は少しさびしい感じになってしまったようだ。まあ，新春以降を期待しよう。



## Traum

M.N.M Softwareが現在開発しているのは，パズルアクションゲーム「Traum」だ。このゲームでは本誌でもお馴染みの寺尾響子さんが，背景のグラフィックおよびキャラクターのデザインを担当している。

背景は立体感を出すためにサイクロンで描かれており，その色づかいやキャラクターのかわいらしさがあいまって，実にほのぼのとした画面に仕上がっている。

ゲームは，スタート地点にいる3匹のネズミをその面のゴールまで誘導してやるのが目的となっている。プレイヤーは上下前後左右に動くことができる妖精を操作して，アワの道を作っていく。ネズミはこの道を通ってゴールを目指すというわけ。で，いろんなキャラがじゃまをする。

詳しいゲーム進行の様子などはまだ不明だが，キャラや仕掛けの動きなどが楽しめるパズルゲームにしたいそうだ。タイトルの「Traum」（トラーム）はドイツ語で「夢」という意味。

X68000用　5″2HD版　価格未定
M.N.M Software　☎0423(60)3084





### ネコも杓子もオーバーテイク

| | | |
|---|---|---|
| 1. | オーバーテイク | 1 |
| 2. | ムーンクレスタ＆テラクレスタ | 10↑ |
| 3. | ふしぎの海のナディア | 6↑ |
| 4. | ファイナルファイト | 2↓ |
| 5. | ポピュラスII | 4↓ |
| 6. | スターウォーズ | −↑ |
| 7. | エトワールプリンセス | 7 |
| 8. | ストライダー飛竜 | 3↓ |
| 9. | グラディウスII | 5↓ |
| 10. | 出たな!! ツインビー | 9↓ |

しかしですね。F1がブームだとか，ズームの久しぶりの作品だとか，X68000にはドライブゲームが少なかったとか，いろいろ理由はあるんですが，なんじゃいねこの得票数は。

2位の得票数のざっと4.5倍。こんな大差はいままでに経験がない。TOP10の歴史の中で文句なし，最高のブッチギリぶり。集計時点では発売になったばかりなので，評価を下しているハガキは少ないのですが，とにもかくにも「ズームだから間違いない！」という期待の声が圧倒的ですね。

2位につけたのは「ムーンクレスタ＆テラクレスタ」。電波新聞社がこんなに上まで上がってきたのは，ずいぶんと久しぶりのような気がします。流行にとらわれず名作をリリースするソフトハウスの姿勢には，高い評価が集まっています。

あれ，「ファイナルファイト」がズルズルと順位を下げてますねえ。ホレ，みんな頑張らないと。アレのこともあるし。「ストライダー飛竜」も同じく急降下。こちらも集計時点ではまだ発売になっていないので，発売後の得票の伸びに期待したいですね。今月は「オーバーテイク」発売に話題をさらわれてしまった感じです。

7位の「エトワールプリンセス」は前回と変わらず踏ん張っていましたが，発売が延び延びになっています。早く発売されないと落ちていってしまうのではないでしょうか。

次点には，「三國志III」と「ロードス島戦記II」。売れ行きそのものはけっして悪くないと思うんですが……。

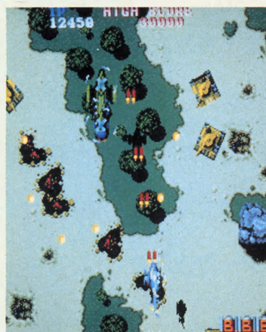
来月は「ストライダー飛竜」が発売されて，「オーバーテイク」を遊びまくった人々の評価が下されます。乞う御期待。（浦）

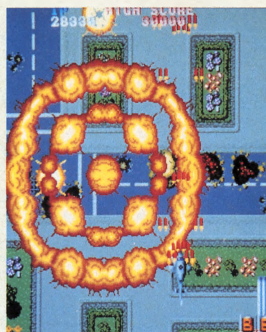


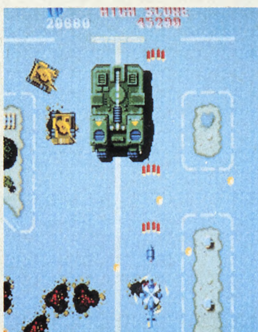
## 究極タイガー

飛んでくる弾を縫うように避けながら，敵を破壊していく。「究極タイガー」はそういう手首の痛くなるゲームだ。地上に並ぶ砲台，物陰や画面の外から不意に現れる戦車，自分と同じく空を飛んでくるヘリコプター。そんなじゃまものどもをひたすら撃ちまくるのである。味方となるのは大型ヘリを破壊すると出現するパワーアップウェポンやボンバー。

X 68000用　3.5/5″2HD版　8,800円(税別)
KANEKO　☎0424(24)7752







## 機甲装神ヴァルカイザー

PC-9801で出ていた，サイレンスの「機甲装神ヴァルカイザー」がX 68000に移植され，TAKERUで発売される。基本的には普通のアドベンチャーゲームであるが，このゲームではヒーローメカが動き回るアニメーション部分に特に力を入れている。トータルで600枚のグラフィックのうち，動画枚数が400枚なのだそうだ。

X 68000用　3.5/5″2HD版5枚組　4,800円(税別)
ブラザー工業(TAKERU)　☎052(824)2493

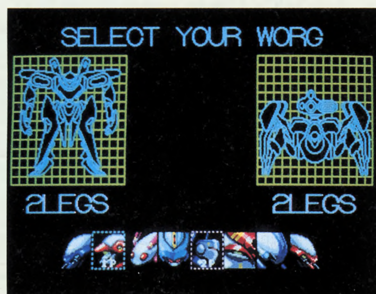


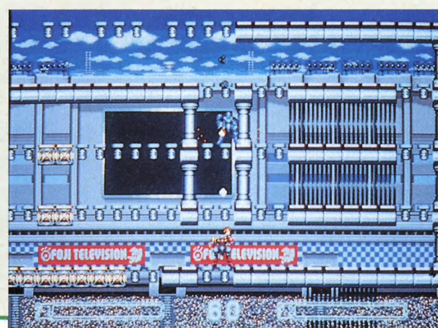



## ストライクレンジ

ロボットどうしが戦うこのゲームの舞台は多層構造。しかし，床には通り抜けることができるものとそうでないものがあるというのがミソ。この床を利用して上下に移動し，敵を狙い撃つ。攻撃は2種類で，ボタン1と2に振り分けられている。攻撃1は威力は弱いが連射可能，攻撃2はその逆ということになる。照準は自動的に行われるので，飛び回って撃ちまくれ。

X 68000用　3.5/5″2HD版　4,800円(税込)
ブラザー工業(TAKERU)　☎052(824)2493

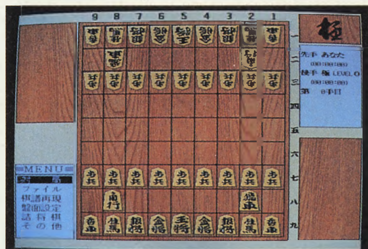


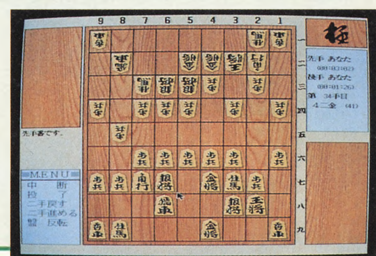




## 極

前回紹介したときはFM TOWNS版の写真であったが，ようやくX 68000版の画面写真が届いたので紹介しておこう。ゲームの内容はほぼ変わらず，多段階のレベル設定，詰め将棋解答モードなどもある。しかし，このテのゲームではCPUスピードの違いがモロに出るので，思考時間は少し長くなる。やむをえないだろう。

X 68000版　5″2HD版　12,800円(税別)
ログ　☎03(3837)2595




## チェイスH.Q.

紹介が遅れてしまったが，「チェイスH.Q.」はすでに発売中となっている。アーケードゲームでご存じの，犯人の車を追って強制停車させるというルールはそのままだが，ゲーム的には少し変わっている。コースの分岐がなくなっていたりということだ。肝心のドライブ部分は若干パワーダウンという感じ。ターボボタンを使ってもあまり速くならないのは残念。

X 68000用　3.5/5″2HD版2枚組　7,800円(税込)
ブラザー工業(TAKERU)　☎052(824)2493









# TREND ANALYSIS

## 1992年10月の月間売り上げベスト10

| POINT | タイトル | 発売元 | 発売日 |
|---|---|---|---|
| 541 | MATIER | サンワード | '92/10/9 |
| 321 | デスブレイド | SPS | '92/10/30 |
| 257 | キャッスルズ | ビクター音楽産業 | '92/10/23 |
| 229 | ふしぎの海のナディア | ガイナックス | '92/10/30 |
| 156 | バーンウェルト | グローディア | '92/10/30 |
| 103 | ポピュラスII | イマジニア | '92/8/28 |
| 91 | バトルテック | ビクター音楽産業 | '92/7/10 |
| 64 | 三國志III | 光栄 | '92/5/28 |
| 53 | シュートレンジ | ビッツー | '92/7/24 |
| 32 | グラディウスII | コナミ | '92/2/7 |



今回はゲームソフトではない「MATIER」が1位ということになった。特に，九十九電機ではゲームソフトの大作並みの本数が売れたそうだ。

この「MATIER」は多彩なエフェクト機能を備えたペイントツールである。読み込める画像ファイルもさまざまなフォーマットに対応しており，レタッチに使用することをメインに置いているようだ。

この機能で39,800円という価格は破格ともいえ，実際のところ，Macintoshなどの100,000円台のソフトと比べても決して見劣りしない。

自分で一から描かなくても，取り込んだ画像などにエフェクトをかけているだけで十分楽しめるので，ふだんあまり絵を描かないという人の購買欲も刺激したのだと思われる。まさにX68000ユーザー必携のソフトというところであろう。

2位から5位は初登場のゲームソフトが占めている。ここでこの4本のソフトの発売日に注目していただきたい。すべて月末となっている。集計は10月1日から31日まででなので，2日間あるいは1週間程度の売り上げのみが計上されているのである。「MATIER」がトップに立ったことには，こういう要素もなんらかの補助となっているだろう。

2位の「デスブレイド」は対戦型格闘ゲームだ。ディスク5枚組というだけあって，対戦相手のグラフィックや攻撃方法は多彩かつ派手である。もともとこのゲームが好きだったという人はもちろん,「ストリートファイターII」の代わりに買ったという人もいるかもしれない。

3位の「キャッスルズ」は海外からの移植作品。経済や外交，敵からの攻撃などを念頭に置きながら，城を築き上げていくというシミュレーションゲームである。頭が痛くならない程度の思考と，職人たちが動き回りながら城が建っていく様子を見られることがウリとなっている。

4位は「ふしぎの海のナディア」。一部で爆発的な人気となった同名テレビアニメーションを，アドベンチャーゲーム化したソフトだ。ゲーム的には前作の「サイレントメビウス」の流れを汲んでおり，ストーリーを追っていくように，すいすいとゲームが進行するようになっている。キャラクターや設定を楽しんでもらおうというしっかりした姿勢が感じられる。

5位はグローディアの作ったアクションゲーム「バーンウェルト」。ストーリー設定の緻密さや，時折挿入されるアニメーションなどに，このソフトハウスの特徴がよく表れている。

というところで“また来月”になるが，次回はズームの「オーバーテイク」が入ってくる。すでに相当の数が売れたようなのでトップに立つことは間違いない。いったいどれほどのポイントを獲得するか，に注目してみよう。

［データ集計協力店］（順不同）
九十九電機本店
J&P（渋谷/町田）
OAシステムプラザ横浜店
P&A
ラオックスGAME館

ウワサのソフトウェア（海外編）

# POPULOUSII THE CHALLENGE GAMES

「ポピュラス」には「プロミストランド」という続編(というかデータディスク）があった。が，グラフィックが変わり難しくなっただけで，本質的にはあまり変わっていなかったので，ほとんど遊ばなかった。

この「POPULOUS II THE CHALLENGE GAMES」を最初に見たとき，ああまたアレかと思った。グラフィックが東洋風，というより“ヘンなニッポン”風のものに変わっており，スモーレスラーなども当然のお約束のように登場する。こりゃ，典型的なイロモノじゃないのか。少なくともCONQUESTモードはそう。

ところが，だ。こいつにはCONQUESTモードのほかに，「CHALLENGE GAME」というとても面白いモードが入っている。ひと言でいえば，「ポピュラスII」の世界に「レミングス」のルールを取り入れたゲームである。ゲームにはとても短い制限時間が設けられていて，この制限時間内に民衆の何％かを生き残らせるというのがプレイヤーの使命である。







例によって最初のほうの面は簡単。民衆を平地に誘導して家を建てさせれば勝ちとか，襲ってくる敵を自陣に侵入する前に殺せば勝ちとか，ちょっと考えれば解けるものばかり。





だが，面が進むと難しくなってくる。たとえば，敵がどかどかと火山を放ってくるので，それを巧みにかわしながら安全な土地まで民衆を誘導するという面があるが，このタイミングが異様にシビアで，ほとんど最適に近い動きを要求される。

どの面に対してもいえることは,「ポピュラスII」の世界を支配する法則をよく理解し，適切な命令や神業を適切なタイミングで使うことが必要だということだ。そういう意味でこのソフトは「ポピュラスII」の達人にはお勧めである。短い制限時間内で100％の力を出し切ることを要求する，密度の高いゲームといえる。

謎の日本情緒もまあ楽しめるが，サウンド周りが全然変わっていないのが残念。(A.T.)

発売元　ELECTRONIC ARTS

ウワサのソフトウェア（海外編）

# VROOM

少し古いが紹介しておきたいゲームがある。このゲームは偉大なるB級作品だ。

いちばん最初のコースを見たときから，そのぶっとびかげんを感じることができる。Fuji Yama Circuit，そこには“富士”という漢字や浮世絵の看板や松の木が道の脇に並び,“日本うんたら”という解読不可能な漢字が書かれたゲートが頭上を通りすぎる。

敵の車や障害物にぶつかると，またびっくり。普通の障害物に当たると大破してタイヤが取れてしまう。が，数秒するとタイヤがいつの間にか復活して走り出すことができる。まあ，これはまだいい。敵車のタイヤなどの低い障害物に当たると，これはたいへん。なんとF1が「ぴょーん」とジャンプしてしまうのである。

というふうに，聞いただけでインチキ臭いのだが，技術的にはすごいことをやっている。写真を見てわかるように疑似3Dレーシングゲームなのだが，スピード感が恐ろしくあるのだ。描画の速さはもちろんのこと，コースの見通しのよさや道の脇に置かれた平面の物体，コースの数カ所に存在するポリゴンの建物，敵の車の動き，エンジン音のよさが功を奏している。

はるか彼方を走っている米粒大の車まで見えたり，ぐねぐねとうねるようにして遠くまでつながっているコース。これこそがF1と納得がいき，ドップラー効果を感じることもできるステレオのエンジン音。さらに，敵車は2枚のスプライト（前輪/後輪部分の2つ）だけを使ってうまく処理し，実に滑らかな動きを実現。こうした割り切りのよさと技術の積み重ねで，アクセラレータがなくてもむちゃくちゃ速いレーシングゲームになっているのだ。

以前紹介した「WORLD CIRCUIT」が純然たるF1シミュレータであったのに対して，この「VROOM」は障害物避けゲームとしてのF1ゲームを突き詰めている。そのため，緊張感では「WORLD CIRCUIT」をすでに超えている。外見的にはチャチに見えても，その奥に隠れている本質部分が優れているのだ。走っているときに気持ちがよければそれでいい，ということを前提に作られたのだろう。抜群に爽快なゲームである。(R.A.)

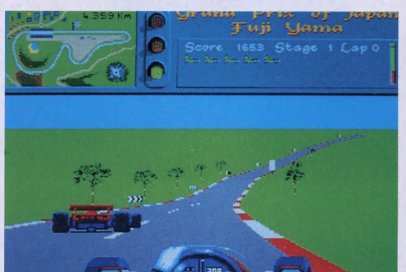
発売元　LANKHOR









# 計算機の中のグランプリサーカス




Tan Akihiko
丹 明彦

**不景気をものともせず爆発的に売れまくっている「オーバーテイク」。1992年一番の話題作となったことは間違いないだろう。シーズン後半になって盛り下がってしまった実際のF1グランプリを，このゲームでもう一度蘇らせよう。**

「オーバーテイク」は，ズームが「ジェノサイド2」以来1年ぶりに放つ，X68000初のF1ゲームである。もうこれだけけで大ヒット作の匂いがプンプンするでしょ。

舞台をF1に移してもズームのパワーは冴えわたる。いつものように圧倒的なグラフィック，音楽，演出。まさしくズームの作る，ズームにしか作れないF1ゲームに仕上がっている。

## FOCA公認ありがたや

「オーバーテイク」を客観的に眺めると，サーキットをラスタースクロールによって表現する，疑似3DのF1ゲームにすぎない。カーレースゲームとしてはごくオーソドックスな作りである。視点はコクピット。これは僕の好みに合う。クリアしやすいという理由で「Virtua Racing」の視点2～4を選ぶ人に僕は美学を感じない。ドライバーズアイが基本だというのが僕の価値観。

ラスタースクロール方式を選んだ以上，ふつうに作っても家庭用ゲーム機やゲームセンターのゲーム機とできることはたいして変わらない。この「オーバーテイク」はデータの質と量と作り込みのこだわりで他を圧倒した。

まず，実在のドライバー，実在のチーム，そして実在のサーキットを使うことで，1992年のF1グランプリ・ワールドチャンピオンシップをシミュレートしている。

まあ，これだけなら現在の状況を見ればごく当たり前の話。ライセンスを活用するといっても，ただ実名を使うだけならお金だけ払えばすむ。が，ゲームを始めてみると，ズームの入れ込みようが，特にグラフィック周りに現れている。

オープニングデモは雰囲気たっぷり。取り込み画像とサウンドをふんだんに使った迫力のシーンがこれでもかとばかりに押し寄せ，気分を盛り上げてくれる。

データベースモードに出てくるポリゴンの車のモデリングもいい。マクラーレンなんかは実車よりかっこよかったりする。

そしてズームの真骨頂は，やはりゲーム本体のグラフィックであろう。

精密に描き込まれた車は見る方向によって別々のグラフィックパターンを使って表示されるので，コーナリング中にも立体感を損なうことがない。たいていの疑似3Dものは，コーナリング中の車のパターンと道路の方向が合っていなくて，立体感が不自然になってしまうのだが，「オーバーテイク」にはそれがない。考えてみるとこれはたいへんなことなのである。

しかも，これらのグラフィックパターンが車ごとに用意されている。カラーリングだけでなく，カウルの形やコクピットのデザインも1つひとつ丁寧に仕上げている。

ただ，あまりにパターンに頼りすぎ，遠近表現が不足している感はある。先行車に追いつくと，がくんがくんという感じで車の表示が大きくなる。こういうところでなら拡大縮小を使ってもいいと思う。

コクピットもかなりの凝りようだ。ハンドルを握る手はハンドル操作やギアチェンジにしたがって本物そっくりの動きを見せる。コーナリング中はハンドルにかかる強大な力を押さえ込もうとするかのような力のこもった動きを見せ，ギアチェンジ時には右手がすっとハンドルを離れてシフトレバーを操作し，すっとハンドルに戻ってくる。そのアニメーションパターンは，1枚1枚手描きで立体感に富み，芸術的ですらある。

いちばん秀逸なのはバックミラー。背景もきっちり映り込むし，後続車も先行車と同様かそれ以上の精密描写で見せる。バックミラーに映った赤白のマシン，変化するコーナーや背景のグラフィック。こんなところまで描き込んであるのだ。

こうしたこだわりは，ゲーム性とは別に，ゲームの雰囲気を盛り上げるのに絶大な威力を発揮する。たとえばバックミラーにセナのマシンが映ると，異様なプレッシャーを感じて，ついつい操縦ミスしてしまったりするのだが，これなどはFOCAのライセンスを取った効果が十分に働いているいい例である。

## 家庭用ならではの

ゲーム構成も僕の好みだ。これはアーケードそのままの移植でなく，家庭用オリジナルのデザインであることがプラスに働いて




X68000用 3.5/5″2HD版5枚組 9,800円(税別)
ズーム ☎011(613)0191



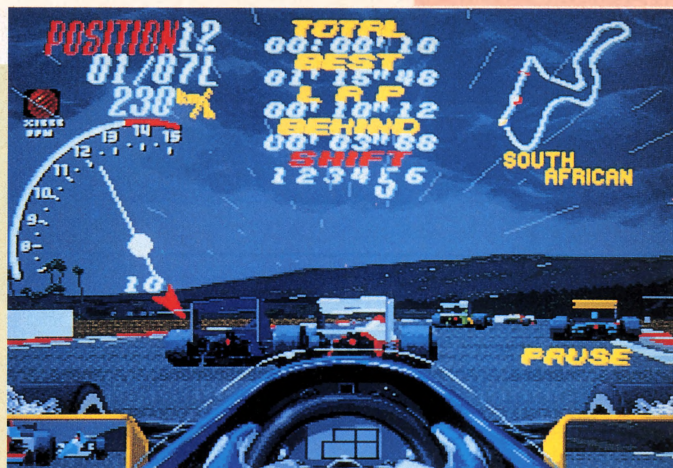

グリーンシグナル点灯。スタートは難しい




超高速コーナーにアクセル全開で突っ込む



バックミラーを見ながら後続車をブロック



ウェットレースは本物より速く走れる

いる。X68000オリジナルのゲームを作る数少ないソフトハウスであるズームのなせる業といえよう。

アーケードのカーレースゲームの何が苦手といって，例の「X秒以内にN周しないとゲームオーバー」というシステムである。僕がゲームセンターに出向いてまでプレイする数少ないゲームのひとつである「Virtua Racing」にしてもこのシステムから逃れることはできない。これはもちろん，1回いくらというアーケード機の制約の中では，当然のデザインではある。

「オーバーテイク」の場合は予選と決勝がきちんと準備されていて，1シーズン全16戦を通して遊べるのはもちろんのこと，2台のX68000を接続して遊ぶ対戦モードや，好きなコースを好きなだけ走れるテストモードまである。レース距離も好きなように設定できる。これはとてもアーケード機には真似のできないことである。

強調しておきたいのは走りのみに集中できるテストモード。サーキットを独占でき，何者にもじゃまされることがない。何周でも納得いくまで走り込む。とりあえずコースアウトやクラッシュをするまで走り，限界を見きわめる。そしてタイムアタック。

さらに，実在のサーキットを再現しているので，自分のラップタイムを現実のコースレコードと比較しながら，タイムアタックに励むという遊び方も可能である。この“永遠のフリー走行”，とてもストイックで面白いと思うのだが，あまり一般的じゃないかな。

こんなプレイはゲーセンでお金を払っていたら，とてもやっていられない。でも，もし「Virtua Racing」にこういうモードがついていたら，通常の5倍の料金がついていてもやってしまうかもしれない。

あと，要望といえば，リプレイ機能とか，コンピュータによる模範走行を見られるモードもほしかったかな。僕はあまり上手じゃないので，見事な走りというのも見てみたいのだ。

## X68000ゲームの手本です

もちろんハードディスクにインストールできる（約4Mバイトを占有する）。キーディスクでコピープロテクトしてはいるが，立ち上げが早い。インストーラの操作性はよく，どのドライブにもインストールでき，どのドライブにインストールしても，システムは自動的にハードディスクを認識してくれる。ここまできっちり作ってもらうと気持ちいい。

当然ながら音楽周りもきっちりとローランドの音源モジュール（MT-32，CM-64，SC-55）に対応。RS-232Cポートに接続するMIDIアダプタにも対応。「オーバーテイク」ではRS-232Cポートが大活躍する（MIDIまたは対戦）。

入力もデジタルはもちろんアナログジョイスティックもサポートしている。

レバーとボタンの割り当ての設定も細かい。カーレースの操作をジョイスティックに割り当てる場合，王道と呼べる割り当て方は存在しない。ハンドルが左右レバーというのは当然としても，アクセル/ブレーキをトリガにするか前後レバーにするかは意見の分かれるところ。だから好みに応じて設定できるのはうれしい。

## 充実の演出

演出に凝るのもズームの伝統。

シャシーが路面をこすって火花が散るというのはめずらしくはないが，あると雰囲気が違う。ウェットコンディション下では雨粒が降り注ぎ，時折稲妻が光る。鈴鹿ではなんと観覧車まで回っている。

効果音も頑張っている。エキゾーストノートに，FM音源とAD PCMの両方を用意している。爆音というのとは違うが，悪くない音である。正直いってエンジンの音をX68000の内蔵音源で表現するのは無理だと思っていたので，これには驚かされた。技術的に見ると興味深い。ちなみにFM音源でプレイしたほうがゲームが重くならない。

音楽はさすがズームといったところ。当然，FM音源，MIDIの両方に対応。オープニングからレースまでたくさんの音楽が用意されている。コースごとにBGMも変わる。あくまでオリジナル曲にこだわったのか，「TRUTH」は入っていない。

特にいいと思ったのは，予選のBGM。



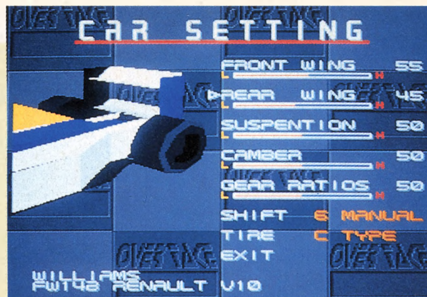
さあ開幕だ。1992年度は南アフリカGPが復活



ジョイスティックは好みに応じて設定できる



ピットインするとポリゴンの人が待っている



セッティング画面。車はポリゴン

タイムアタックに突入する瞬間に演奏を開始し，ドライバーの高揚した気分を盛り上げる。

ご存じのように，F1の予選ではピットアウトしてからの1周を流して走り，その次の1周を全力で走り，タイムを計測する。ホームストレートでアクセルを全開にし，タイムアタックに突入する。この瞬間は，ドライバーのテンションがピークに達する瞬間なのである。その瞬間に絶妙のタイミングでBGM，このあたりの演出は心憎い。

## さあ，走ろうか

「オーバーテイク」の走行感覚そのものは，現実とはかなり異なる。メガドライブの「スーパーモナコGP」を意識しているといっていいだろう。

先ほども述べたとおり，ラスタースクロールでサーキットを表現しているため，その限界からか，ちゃんとシミュレーションできているとはいえない。

とりあえずヘアピンや直角コーナーは表現できない，または非常に難しい。テクニックとデザインしだいではある程度カバーできるが，ラスタースクロールタイプのレースゲームは徹底して爽快感を追求するのが得策である。

つまり，このゲームではコーナリングにおいて，シフトダウンやブレーキングは重要でない。タイミングさえよければ，鈴鹿をアクセル全開のまま5～6速で周回できる。ヘアピンでも5速まで落とせば十分間に合う。シケインはうまくすれば，6速のままで回れる。これは，シミュレーション性を追求してゲーム性を落とすよりは爽快感を優先しようとしたのだと好意的に解釈しておく。

僕は6速オートマチックを選ぶ。オートマチックでも手動で強制的にギアチェンジできるので，ふだんはギアチェンジを機械に任せておいて，コーナリング時に手でシフトすることができる。自動のオーバーレブ防止装置を備えているようなものだ。

道に沿って走っていれば，それなりに周回できてしまう。ただしタイムを出そうとするなら多少の努力は必要だ。コーナーが見えてからあわててハンドルを切っても回り切れないコーナーもあるし，きっちり減速して5速に落とさないと回れないコーナーもある。これを把握せずに漫然と走っていると，次々と押し寄せてくるコーナーに翻弄されてしまうのだ。

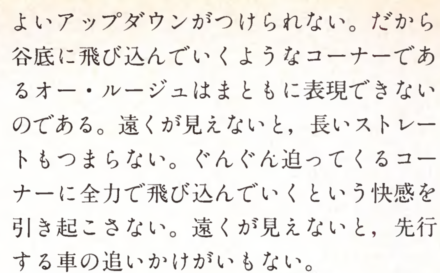
少しずつでもコースを覚え，要所要所できっちりと締めるのが，ラップタイム向上への道といえる。目で見て走ってるうちはまだヒヨコだ。

ヘアピンやシケインとは異なり，高速コーナーは比較的よくできている。鈴鹿の130Rやシルバーストーンのブリッジコーナーは，もともとタイミングをきっちり合わせてアクセル全開で飛び込んでいくタイプのコーナーで，「オーバーテイク」ではよく再現されている。

また，長くて半径の大きなコーナーも得意。エストリルの最終コーナーなどがそうだが，数秒の間アクセルを全開にしたまま横Gと闘うタイプのコーナーの雰囲気もいい。要するに「オーバーテイク」はアクセル全開が自然な状態であるというコーナーを得意にしているといえる。

しかし，全体的にコースの作りは車のグラフィックに注ぎ込んだ情熱に比べると落ちる。特に，モナコのトンネルやスパのオー・ルージュには幻滅させられた。

コースの見通しが妙に悪い。つまり遠くが見えない。遠くが見えないので気持ちの



よいアップダウンがつけられない。だから谷底に飛び込んでいくようなコーナーであるオー・ルージュはまともに表現できないのである。遠くが見えないと，長いストレートもつまらない。ぐんぐん迫ってくるコーナーに全力で飛び込んでいくという快感を引き起こさない。遠くが見えないと，先行する車の追いかけがいもない。

スピード感のない理由はひとつは遠くが見えないため，もうひとつは画面があまり動かないためである。背景はよく描き込んであるが，スクロールする1枚絵にすぎない。ストレートを走っている間はほとんど静止画。コース沿いに近景となる物体を配置することは，確立されたノウハウであるが，壁がその役に立っていないのも不思議だ。あと，路面が単調なのもマイナス。ウソでもいいからタイヤのスリップマークを入れて，それを高速で流すくらいのことはすべきではないかな。

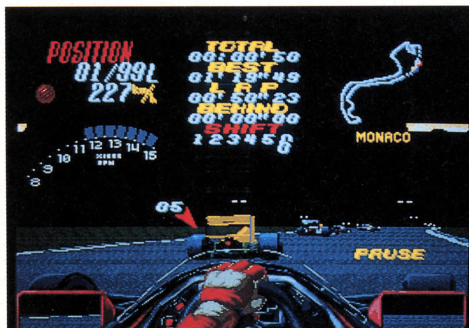
スピード感のないこともあいまって，ストレートがとにかく気持ちよくない。一般にF1のコースは抜きどころが少なく，モナコに至っては事実上トンネルのみである。その1周に1度のチャンスに向けて，先行車にくらいついて間隔を詰め，スリップについてチャンスを窺い，トンネルの中でサイドバイサイドに持ち込み，シケインの突っ込みで抜き去る。ほかのサーキットも同様。最もスピードの伸びるストレートで豪快にオーバーテイクする，これがF1のレースにおける至福の瞬間であるはずなのだ。

「オーバーテイク」のオーバーテイクは，どちらかというと先行車を乗り越えるイメージ。トラックで轢きつぶすみたいで，F1とは別もののようだ。

## 辻褄合わせのラップタイム

全区間を最高速で走るわりにはラップタイムが比較的リアル。コースの長さを上手に調整してあるのだ。しかし，唯一それが破綻するのがウェットレースである。

晴れならドライ，雨ならウェット。レー



両脇の草地が興ざめなモナコのトンネル



カーブでの追い越しもそんなに難しくない

スモードでは1992年のレース当日の天気に設定できる。きっと，このために発売を1992年シーズン終了まで待ったに違いない。

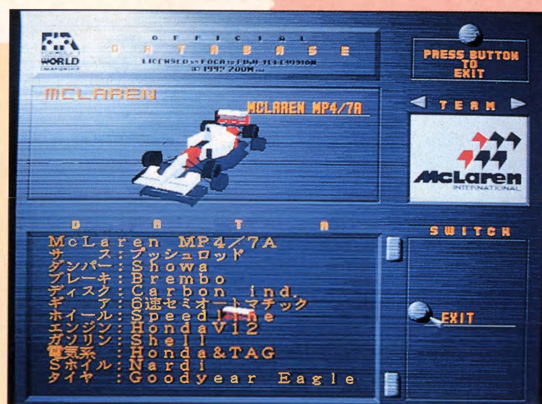
ここでいいたいのは，ウェットレースを走ると「オーバーテイク」のシミュレーションとしての忠実度がわかるということ。

その理由はウェットコンディションだと簡単にポールポジションが取れることにある。現実の予選タイムはドライコンディションの10〜20秒落ちなので，「オーバーテイク」の予選も，そうなるように自分以外の車のラップタイムをもっともらしく計算している。だが，実際に走ればわかるが，ちょっと練習すればドライコンディションの5秒落ちくらいで楽に周回できる。だから結果として，ぶっちぎりのポールポジションになる。そして決勝では，コンピュータの車のラップタイムはきちんと予選タイムを上回るのである。

このことは，図らずもウェットコンディションのラップタイムの辻褄合わせがうまくいっていないことと，コンピュータの車が実際に予選を走ってタイムを計っているわけではないことを露呈している。



実在のドライバーから選択できる



データベースモード。パネルの質感がいい

## タイムアタッカーの気持ち

画面表示のレイアウトはいいと思うが，タイム関係は工夫の余地があったと思う。F1にはいろんな種類のラップタイムがあるが，どのタイムを表示してどのタイムを省略するかがまったく練られていないという印象を受ける。

たとえば「BEST」にはコースレコードでなくそのセッションのベストラップを表示すべきである。レース中にファステストラップがわからないのは変。また，それとは別に，自己ベストもほしいところ。予選のタイムアタックの参考にもなるから。

また，「BEHIND」はなぜ1位の車からのタイム差なのだろう。すぐ前の車との差のほうが情報として面白いのに。1位になると表示が0秒になるというのもつまらない。2位との差にすべきである。前の車に追い付き追い越すのがレースの醍醐味というのであれば，すぐ近くの車とのタイム差をドライバーに教えるのが正しいデザインというものであろう。

ラップタイムが見にくくなっている（たぶん色が悪い）のも難点。書体のデザインに凝るのはいいが，タイムアタックしているときには区間タイムも知りたいから，一瞥してわかるものであってほしい。

## 現実に引き戻される

ピットイン関係は完全に浮いている。本コースとピットの切り替えにブラックアウトするというのは興ざめだし，わざわざポリゴンを使うのも理解できない。

こういうところには，ゲーム処理を全部ポリゴンでやろうとしていたあとが窺えるが，X68000にポリゴンは荷が重すぎたのだろうか？　認めたくない……。

確かにポリゴンは処理速度では圧倒的に劣る。しかし，技術とノウハウしだいでかなりいいセンいくことも事実。このへんは西洋人にはまったくかなわない。

コースわきのタイヤバリアはけっこう迷惑。あれにぶつかると止まってしまうのでいらつく。現実のレースだとクラッシュに相当するから，走れるだけましともいえるが，あれのせいでコースアウトしないようにしないようにという消極的な走りになってしまった。

## おわりに

異論があるかもしれないが「オーバーテイク」は緻密なデータを武器にした，“参加できる公式F1ガイド”といえる。1993年の各チームの体制とグランプリ開催スケジュールが確定したら，1993年版データディスクを出すといいかもしれない。疑似的にでも運転した体験をもっていると，レース観戦も数倍面白くなるものなのだ。

国内の家庭用F1ゲームでは，「実在のデータを緻密に再現する」タイプがトレンドとなっている。「オーバーテイク」はその中でも最もレベルが高い。少なくともカタログスペック上は完璧。

あとは「精密なシミュレーション」と，「爽快なドライブ感」を実現したタイプが出てくれれば，何もいうことはない。この2つに関しては，厳しいことをいうようだが，「世界のレベル」を知ってほしいのだ。

フジテレビはF1グランプリのゲーム化権を独占している。現在，このためにF1ゲームが移植されたり，開発されることが難しくなっている。たくさん売れるコンシューマ向けとは違い，市場の限られたパソコンゲームでは莫大なライセンス料は大きな問題になってのしかかってくる。

ここまで僕がF1のコースを見てきたようなことをいえたのは海外のとあるF1ゲームのおかげであるが，もしいま述べたような問題が国内機種向けに移植される障壁になるとしたら，たいへんな悲劇である。

「オーバーテイク」はライセンス問題をクリアして世に出てきたものだ。そして，その決断は賞賛に値する。過大な時間と費用の浪費させられたことであろう。が，そんなにまでして「オフィシャル」を取らないと発売できないっていうのはやっぱり変だ。フジテレビさんにはゲーム化権とテレビの放映権を一緒にしないでもらいたかった。いいかげん古舘アナも引っ込めてほしい（支離滅裂）。

### 売れることは保証します

「オーバーテイク」は今年最大の話題作といっていい。まず1年ぶりのズームの作品，しかもX68000初のF1ゲーム，さらにFOCAのライセンスを背景にした実在のデータ，徹底した作り込み，雰囲気たっぷりの演出。話題性もヒットする要素も十分である。実際，飛ぶように売れていると聞く。X68000用ゲームソフトの動向をつかむためには押さえておくべき1本である。

| 総合評価 | 0　5　10 |
|---|---|
| 音楽 | ★★★★★★★★★ |
| グラフィック（静） | ★★★★★★★★★ |
| グラフィック（動） | ★★★★★ |
| 演出 | ★★★★★★★★★ |
| 爽快感 | ★★★★★★ |
| バックミラー | ★★★★★★★★★ |
| ピットイン | ★★★★★ |

# 無敵のハイパースタント

Yokouchi Takeshi
横内 威至

カプコンのX68000ゲーム第2弾，「ストライダー飛竜」が発売された。少し前のアーケードゲームからの移植ということになるけれど，キャラクターの動きの多彩さなどはいまでも色褪せていない。「ハァッ！」の叫び声も健在だ。




ついに出てしまった。私はかなり以前からこのゲームの移植を望んでいたが，ハード的に無理があってだめなものかなあと落胆していたのである。しかし，その壁を超えて，「ストライダー飛竜」がここに現れたのだ。

このゲームはいわずと知れたカプコンの名作アクションゲームだ。当時，そこいらのアクションゲームに白い目を向けていたような者をも驚嘆させた，革命的なアクションゲームなのである。

結局はシューティングゲームの戦闘機が人間になっただけにすぎず，流れにそって敵を撃ち落としていくようなパターン性の高いアクションゲームばかり並ぶなか，このゲームは自由度の極めて高い見事なアクションをメインとしていた。

とはいえ，予想したほどのスマッシュヒットとはならなかったのだが，それはマニアックな攻略，点数稼ぎの絶妙な奥行きというものを，このゲームはもっていなかったからである。しかし，それは「ストライダー飛竜」が真にアミューズメントに徹していたためなのだ。

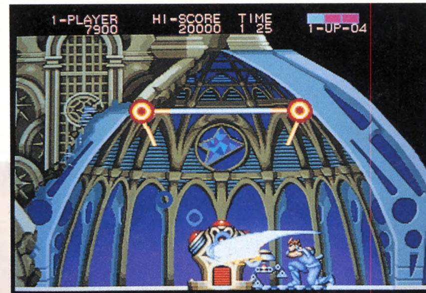
## ストライダー飛竜のススメ

こいつは並のアクションゲームとはわけが違う。なんてったってこれを超えるアクションゲームはいまだにありえないとまでいわれているシロモノだ。ダイナミックなアクション，いや，スタントと呼ぶにふさわしい動作，展開，これこそこのゲームの売り。ほかでは味わえない，アクロバティックなスタントシーンの連発である。そこらのチンケなちまちましたアクションなんかもう見られたもんじゃないよ。

おっと，アクションゲームは好きじゃないって人もこいつだけはやってみなよ。きっと惚れ込むに違いないぜ。

よくある，あのアクションなどと呼ぶにはとても及ばない1ドット調整，タイミング調整などの類のしみったれた操作などは存在しない。「究極の1ドットジャンピングアクション」だとか「垂直同期スーパータイミングアクション」だとか「1ドットのエクスタシー」(あ，知らないか。これはちょっと違うね）なんて，俺はアクションゲームよりもイヤラシゲーム（アダルトとはいっていない）に分類したいぐらいだ。あゝゆうのは非常によろしくない。なぜって，ゲームであることを悪用したイヤラシい要求だからだ。

だって，「私はいつでもどこでも坂でも秒速1.21メートルで動き，水平に5.43メートルぴったりでジャンプして行動します」というやつなんているものか。ゴルゴ13じゃあるまいし。まあ，そういうのを完璧にコントロールする超人的な野郎もいるだろう。それもいいことはいいが，時と場合だ。正確な間合いなんかを重要視することが必要なこともあるけど，それは緊張感であり爽快感とは別のものだ。



ちょっとわかりにくいけど反射レーザー

「ストライダー飛竜」がほかと一線を画している点はここにある。純粋にこの爽快感，躍動感を追求したノリを徹底して，かつてありえなかった新しいアクションの活路を開いたのである。映画でいうならかつてのあのブルース・リー，そんなところだ（反論も多数あろうが)。音楽ならユーロ，あ，でも中身がないなんていわれるから，こりゃやめとく。

## ウハウハダンス小僧

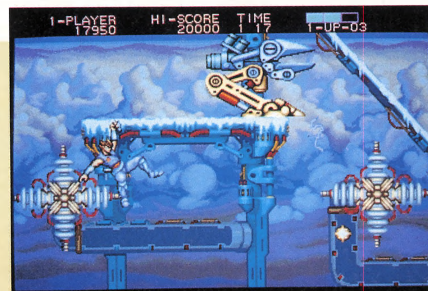
アクションゲームだから，どんどん進んでいき，最後のボス，グランドマスターを倒せば終わりだ。でも，歩いていくだけでは壁に当たるのがオチとなるのが世の常である。飛竜を動かして，というより踊らせていかねばならない。アクションは実に多彩なのだが，1レバー＋2ボタンというジョイスティックの基本形だけで踊らせることができる。



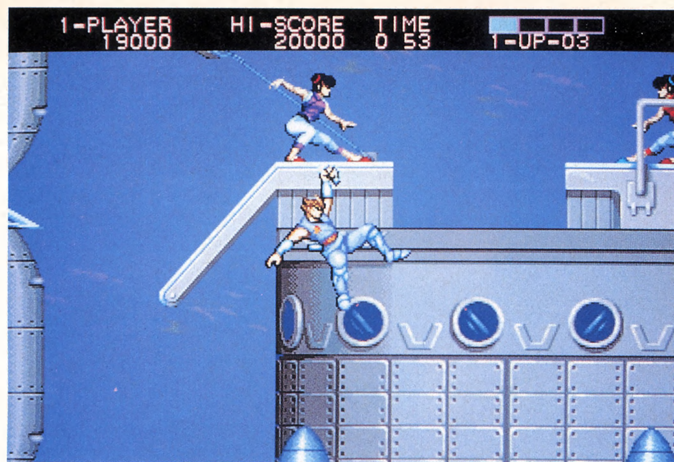

X68000用　5"2HD版3枚組　9,800円（税別）
カプコン　☎03(3340)0718



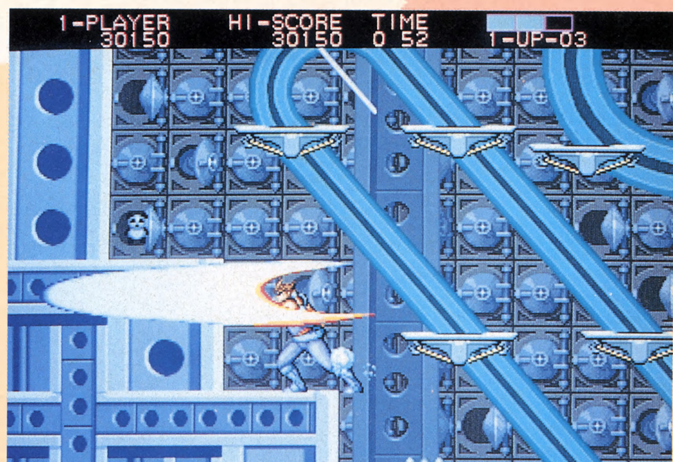
ソロ。つきまとってくる寂しい男



怪しげな発電地帯。後ろが死角だ



飛行艇の娘たち。左の斜面上が安全



謎の隠れキャラ，パンダ。効果は？

壁につかまったりだとか加速したりだとかいう，以前のゲームでは不可能であった欲求を見事に実現してくれたのはこのゲームが初めてだったんじゃないだろうか。かといって操作は単純で，かぎりなく軽快なプレイをまっとうできるのだ。

基本的にレバーの左右で歩くが，下り坂ではグングン加速するし上りでは減速する。関係ないがこいつはすごい走りを見せる。斜面でも体がほぼ垂直な状態のままで走るのだ。私も昔試してみたがあまりの加速についていけなかった。ああ恥ずかしい。そんなことはまあいい。足場の下につかまっているときでも左右にがんばって動く。上にやればスルッと上がるし，下にやれば手を離して降りる。また，壁は登ったり降りたりできる。しかし，全体的に坂だとかばかりで分類してたらきりがない。とにかくレバーを入れたとおりに快く反応する。

ボタンのひとつは攻撃だ。「ハァッ！」って気合いを入れて，サイファーっていう，いわゆる剣をズバズバ振り回す。ノリがいいから，連射にマッチして振ってくれる。どんな状態にあっても振ってくれるし，宙を舞うときでも体をひねりながら左右に斬りつける。しかし，どこかにつかまっていて片手しか空いていないときは連射できない。やっぱりバランスをとるのがたいへんだからだろうね。



危険だが楽しめるカタパルト

もう一方のボタンはもちろんジャンプ。わざわざ１回転しながら宙を舞うあたりはダンス小僧の真髄だ。魅せてくれるねえ。片手でどこかにぶらさがっていてもそこから跳んでくれる，素晴らしき腕力をもっている。しゃがんでいるときに押すとわざわざくるっと回ってスライディングだ（ブレイクダンスではない）。ただし斜面の緩いところでしか滑らない。スピードもあり，滑る距離も長いので歩くより速い。また，爪先も鋭いらしく，これでも敵を倒せる。ブッチャーの靴よりも鋭いらしい。

ジャンプは結構気持ちいい。ほかのゲームではなんだか自分が重たそうなのとか，あるいは妙に軽々しいのなんかがあるけど，こいつはきれいにスーッと伸びるように宙を舞ってくれる。加速時の跳躍なんて実に伸びがよく，なかなかに優雅な気分である。ゲームの躍動感を高めている要素のひとつになっている。

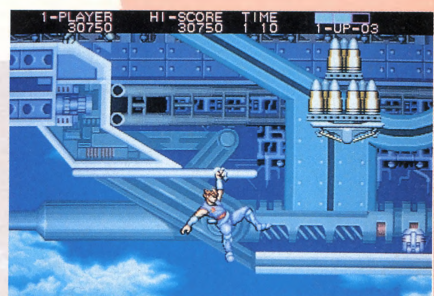
ということで，動きの多彩さ，パターンの多さは見事なことがわかっていただけただろうか。敵だって坂に沿って歩いたりして，がんばってくれている。メモリに感謝してしまう。

## 3つのしもべに命令だ

さて，もちろんパワーアップというか，役立つアイテムなんかもある。道に落ちて



逆重力地帯。空中砲台は厳しい



空中を伝っていくときはかなり恐ろしい

いたり，荷物運搬用の変なロボットが運んでくるカプセルに入っている。

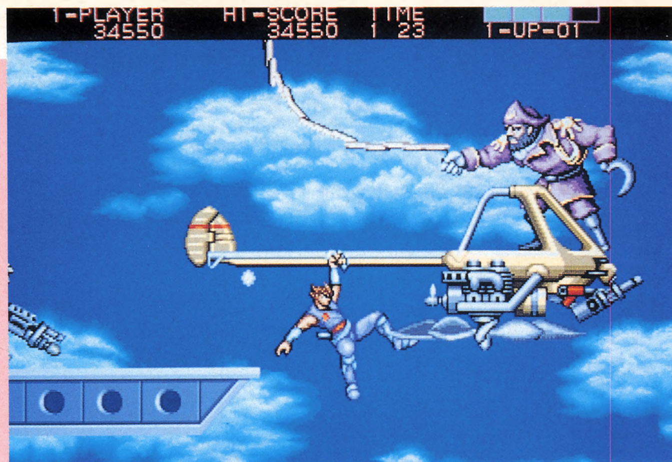
基本的なのはサイファーの攻撃範囲が大きくなるアイテムだ。時間ではなく振った回数で効果が消えてしまう。そして，強力な味方が，円盤型２足歩行ロボットと呼ばれるオプション君だ。飛竜を中心に適当に動き回って敵に体当たりして殺す。変なビームを撃ってもくれる。こいつは２つまで取れるが，ひとつ取るごとにライフが１目盛減ってしまう。で，そのあとにダメージを受けたりして，ライフが取ったときの残りの値よりも少なくなると消えてしまうのだ。いつまでも残したいならこれが出るところまでにダメージを受けておいて，取ってから体力を全回復させればよい。

このオプション君は，左右方向ならある程度レバーで制御可能で，動かすのに慣れてしまえば，もうほとんど無敵状態だ。本当に強いぞ。

そして，こいつが２つあるときにはさらに新しいオプション君が使える。豹型オプションだ。これを取ると，前のオプション君はいなくなってしまう。攻撃力はあるらしいがもったいぶってあまり動かない。そのうえ飛竜を盾にして先を歩かせるような図太いヤツなので，あんまり使いものにならない。でも，私はオプションは強すぎて嫌いだからこいつをよく使う。ダメージを




重力制御室。届かないのがはがゆい




部下を盾に逃げようと奔走する愚かな船長

受けるか時間がたつと元に戻る。

あと，特定のオプションパックで鳥型ロボットも登場するが，これもあまり役には立たない。

白く光っているアイテムは無敵＋トレースだ。自分のあとを，飛竜の残像２人がトレースしてくれる。結構気持ちいい。

また，「飛」マークは体力回復，「竜」マークは体力ゲージアップだ。あとは飛竜人形で１人増える。

## 夜遊びガイド

踊るための舞台は当然それらしく派手なものとなっている。六本木から始まってウォーターフロントを通って芝浦を通過，ではない。赤い国の都市をスタートにシベリアを走破。空中戦艦ってやつの中で暴れて，アマゾンのハーレムへ。そして最終目的地，サードムーンへと走り抜ける。

始まりから終わりまで，スキのない激しい展開に一瞬の油断も許されないのだ。でも，現段階のバージョンでは４面のアマゾンまでしか入っていないから，究極の見せ場は味わっていない。残念。製品版を心待ちにしている。

ステージは全５面だが，各面はもちろんのこと，１歩先でさえもう未知の世界，常に違った展開が用意されている。いやあ，改めて見てもこれには感心してしまう。なんというか，アクションゲームというと，わりと似通った風景を似通ったパターンで似通った気分で進むのが当たり前であったが，これは本当によく練ってある。ひとつの面でもいろんな舞台があり，全体としてただならぬ広がりを感じるのだ。

最近はハデなイベント，ハデな画面っていうと，安易なゲームではすぐ拡大縮小だとか，ラスタースクロールなんかで，ただ画面効果に訴えがちである。しかし，そういうのは別にゲーム自体の流れに影響を及ぼすわけではなく，私にいわせると邪道である。そういうゲームは正しいイベントの姿をこの「ストライダー飛竜」から学ばねばならない。次々と移り変わる激しい舞台こそ真のイベントであり，展開が直接イベントであるようでなければ優秀なゲーム構成とはいえないであろう。

私事ではあるが，「グラディウス」シリーズはそういう点で非常に印象深い。分析してみると，展開に重点があると思う。背景が最も印象深いのは当然ではあるが，その背景こそが重要な敵であった。つまり背景が生きているのである。背景が背景である以上に舞台なのである。ここがよくできたモノのポイントなのだろうか。

そこまで作り込まれたゲームは，キャラクターや背景をキャラクタ文字で置き換えても素直に楽しめるはずであり，真に必要なものが何かは明白であろう。単なるラスタースクロールなんて子供だましにすぎぬイベントであるのだ。

## 展開は飛竜のために

さて，「ストライダー飛竜」に戻るが，これはかなり練られた展開を見せてくれる。展開とアクションの多彩さのために，道は１本ではあるが抜け方というのは人それぞれに委ねられている。




ピラニアだがノーダメージ。50点かせぎ

つまり，だ。アクション系によくある攻略の「ここに乗って斜め上に撃ってあいつを殺してから次にこうしてああして」って感じのやつはあまりない。おおまかな抜け方だとか特定の敵の倒し方なんかはあるけど，このゲームにおいてそんなものは無用なのだ。なんといってもこのゲームの目的はいかに美しくプレイするかであり，それが爽快感のうえで重要だと思う。ハデに動き回って，ハデに見せ場を作り上げていくべきである。

さて，簡単に遊び場を巡ってみようか。

### 1発目　カザフ共和国

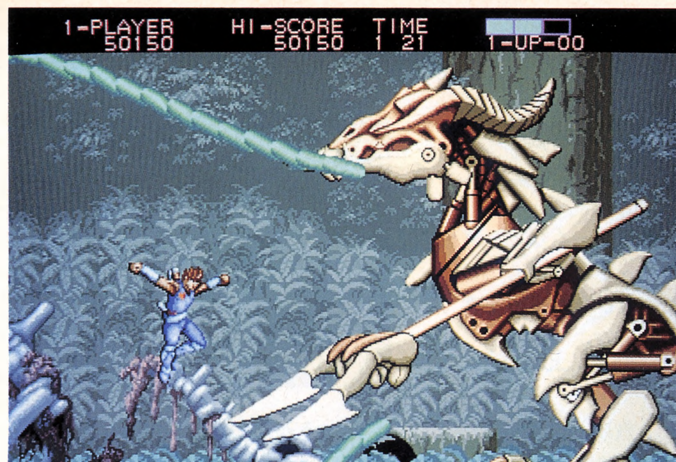
情けない兵隊たちが歓迎してくれる。こいつらは腰抜けだ。なかには近づくとあわてて逃げ帰る愚かなやつもいて泣かせてくれる。モスク護衛用の無能小型兵器ラスカルなんかもいる。たまにこいつに遊んでもらうとこれまた情けない攻撃だ。

少し進むとドーピング男の部屋だ。無視してもかまわない。さらに進んで満月のモスクを越えていく。ここで戻っていって加速ジャンプ遊びをしてもよろしい。なぜかモスク警備をしている専用ロボットもいるが敵ではない。叩き斬って進むのだ。

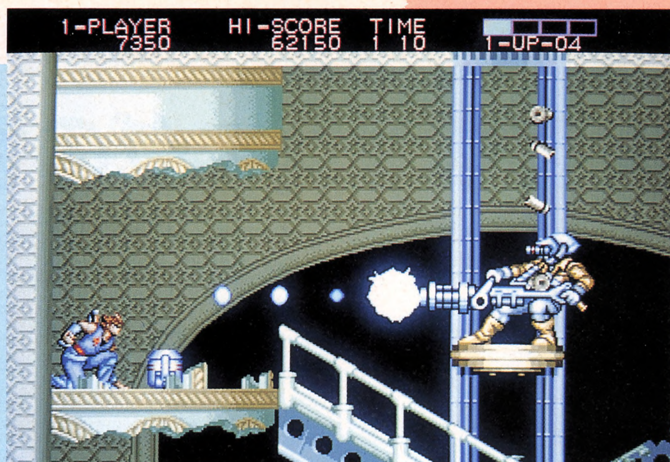
モスクの中に入るとレーザー反応装置があって反射レーザーの応酬だ。メーターがついているあたりがシャレている。切り崩して降りるとマシンガン男，飛び散る薬莢




恐竜地帯はつらい。無敵を探せ！



結構手ごわいやつ。下をくぐって右には……



マシンガン男だ。強そうだが一撃で死ぬ

がカッコイイ。ラストは搾取政治家たちが群がって合体するムカデだ。ムカデも上に乗ったりして遊べるから，適当に遊んでやるとよい。

**2発目　シベリア走破**

舞台はシベリアへ。遠吠えする狼たちを斬り刻んで秘密工場へ進む。ここはゴリラマシーンなんかを作っているクレージーな工場だ。１面で出てきたヘボいマシーンの開発スタッフに，グランドマスターが激怒してシベリア強制送還したらしい。しかし，あいかわらずくだらないマシーンしか作っていない。上まで上がっていくと崖の上。賞金稼ぎのソロと対決し，ゲーム最大の見せ場でもある地雷斜面駆け降り＋ジャンプだ。この快感はやらずにはわからない。ああ気持ちいい。発電地帯に移り，いよいよ空中戦が始まる。

敵のヘリを奪い，はるか上空から降ってくる爆弾をかわしつつ，乗りついで飛行艇まで上昇するのだ。飛行艇に乗り込めば，中国娘３人衆だ。でも弱いから安心を。飛行艇を奪った飛竜は，いよいよすべてが見せ場ともいえる空中戦艦へ。

**3発目　空中戦艦バルログ**

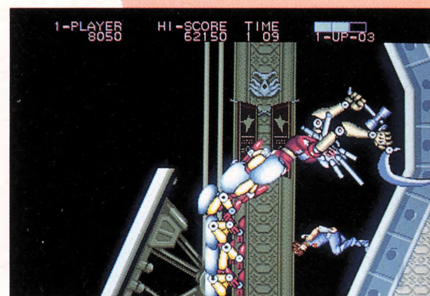
乗り込むといきなり甲板だ。砲台を破壊して内部へ潜入。謎のベルトコンベアを越えていくと迫りくる壁。急いで壁を登れ。裏の部屋からは，なにやら敵がヘリでついてくる。途中から左右に跳びはねながら上がると艦橋裏で敵と対決。

ここには発射カタパルトがあるが，使用は自由なので急いでいるなら飛んでいってもかまわない。保証はしないが。さらに進み，いよいよ心臓部へ向かう。

通路を進むとなにやら-Gだとか書いてある。心臓部には重力コントローラがあるのだ。コントローラを破壊するためにメインルームに入ると，胡散臭い機械にぐるぐる回されてしまう。破壊したらすぐに脱出する。火の手が追ってくるなら，最後部のヘリへ。アホな船長も部下を払いのけてヘリに乗っている。船長を倒してヘリを奪うと南米に向かうことになる。

**4発目　ハーレムでウハウハアマゾン**

場面はアマゾンへ。前半はグニャグニャの蔓でアマゾネスのハーレム地帯である。落ちるとアマゾン川らしくピラニアに食われてしまう。適当に進むとちょっとした崖。インディって感じで巨大な岩が背に迫る。途中の蔓に飛びついて先へ向かうと，そこにはグランドマスターが復活させたらしい恐竜たちがいる。でかいやつの背中に乗って進んでもいいし地面を走破してもいい。ちなみにサイは背中に乗れるから，運んでもらってもOKだ。



ムカデはちょっと強い。でも遊べるやつだ

最後には，機械で作られた恐竜が待ち受けている。

## あとは自分の目で

と，今回はここまでである。最終面は敵基地で，竹槍逆重力滑降，重力コントローラ，強力反射レーザー，ムカデの背に乗って空中を移動，そして……ということになっているのだが，それは製品版で確かめてもらいたい。

ざっとゲームについて流してみたが，このゲームのもつスピード感，躍動感，そしてノリは，やらずに知るには無理があると思う。ぜひとも多くの人に，このゲームを実際に味わってみてほしい。行き詰まりつつあるアクションゲームに，まだまだ奥深き未来があることを痛感させられることだろう。逆流するアドレナリンに，もう君はじっとしていられない……。

### 気合い十分だ，ハアッ!

気になる移植の出来だが，合格であろう。重要なノリは崩しておらず，アーケードそのままの気分で楽しめるはずだ。ただ，やはりスプライトオーバーによるちらつきは出てしまう。一部グラフィック書き換えをしているときや，高速スクロールで処理落ちしているのももったいないが，これはしかたがない。X 68000の能力は十分に引き出されている。よくぞここまでできたものである。「ファイナルファイト」の人数落ちと違ってゲーム自体の損失はなく，かなりハイレベルの移植だ。

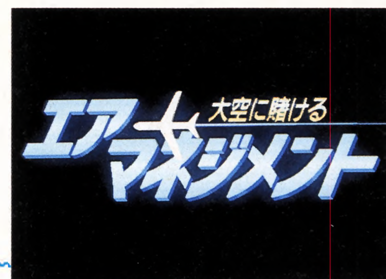
メガドライブ版と比べるとちょっと心苦しい点があるかもしれないが，メガドライブ版はいまではあまり覚えていないからはっきりしたことはいえない。もちろん，X 68000のソフトとしては最高レベルであることは違いない。

さて気になる点は次の作品だろう。やはり，皆が考えているとおりなのだろうか。カプコンに触発されて，ほかのメーカーもがんばってほしい。コナミも次は「魂斗羅」関係を攻めてほしいな。

さあ，もういまはこのゲームしかない。とにかくやってみな。そのうち，飛竜の気合いの「ハァッ！」という声が，脳裏に悦楽の響きとして聞こえ出すに違いない。

| 総合評価 | 0　5　10 |
|---|---|
| ゲーム性 | ★★★★★★★★★ |
| 操作性 | ★★★★★★★★ |
| グラフィック | ★★★★★★★★ |
| 気合い | ★★★★★★★★★ |
| 次回作 | ★★★★★★★★★★ |

# 大空もヤンエグの香りだ




Urakawa Hiroyuki
浦川 博之

**戦争や事故，オリンピックの開催や旅行ブームなど，激変する政情や乗客の動向。航空会社のトップに立つものは，素早く的確な判断を下さなければならない。全世界を覆うネットワーク，それがこのゲームの舞台だ。**

今年の就職戦線は不景気のあおりをくって，長引く傾向にあるそうだ。戦線に終止符を打つコツは，とにもかくにも企業にアポイントを取ることである。

プルルルルル。

「はい，ジャパンエアー人事部です」

「あの，羽田大学のうりゃかわと申しますが，御社の採用のご予定は……」

「社長の椅子なら空いてますが」

「しまった。またこのパターンか」

再び私こと，うりゃかわ君が，光栄のビジネスシミュレーションゲームの世界に飛び込むことになった。前作「リーディングカンパニー」を遊んだ経験からいろいろと考えたこともあるので，今回はそこらへんもからめて紹介していこう。

## 就職先は顔を見て選べ

1992年の就職活動にもかかわらず，社長に就任するとそこは1983年4月だ。うりゃかわ君はジャパンエアー（以下JPA）本社に通される。

**副社長**「ようこそ社長。私があなたの補佐を務めます副社長ナニガシです」

**うりゃかわ**（以下**うりゃ**）「は，どうも」

副社長の顔をまじまじと見つめる。360度どこから見てもヒゲづらのおっさん。

**うりゃ**「JPAはスチュワーデスがきれいだっていうから電話したのに」

そんなこといってるから就職が決まらないんだと小言のひとつもいいたくなるが，ほかの航空会社には美人の副社長もいる。

このゲームでは空港のある都市が全部で22あって，どの都市にも航空会社がひとつずつあるのだ。東京ならジャパンエアー，パリならエアーフランス，モスクワにはアエロロシア。プレイヤーはそのなかから自分の会社を自由に選ぶことができる。香港，ナイロビ，リオデジャネイロなど現実にはメジャーエアラインの存在しない都市を選んでも，エアーサバンナとかナスカエアーとか，怪しい会社をデッチ上げてくれる。

部下の顔も地域によって何種類もあり，なかには女性が副社長の会社もあるというわけ。都市ごとにどんな会社が出てきてどんな人々が登場するのかを眺める。これがなかなか面白くて，私はしばらくゲームを忘れて本社都市を選びつづけていた。

## 航空会社のイロハ

**副社長**「社長の使命は世界22都市にまたがる航路網を完成させ，年間乗降客数3,500,000人を突破させることです。赤字が2年間続くと倒産ですから気をつけて」

**うりゃ**「はいはい。いつものパターンだ」

メイン画面には，世界地図がどーんと構えている。1983年という時代にもかかわらず，世界の航路網はゼロ。プレイヤーを含めた4つの航空会社は一から経営をスタートする。

**うりゃ**「えーと。とりあえずどうすればいいんでしょうか？」

**副社長**「航路を開くには空港の使用権（スロットという）を買わなければいけません。部下を送ってスロット確保の交渉にあたらせてください」

**うりゃ**「で，どこに航路を開くのがいいんでしょうか？」

**副社長**「会議というコマンドがあるので，そこで話し合ってください」

シミュレーションゲームで勝つには出だしが肝心である。そのくせ出だしのときにはルールが飲み込めてないので，どうしたらいいのかわからないことが多いのだ。その点，「エアーマネジメント」では会議を開いて部下の意見を聞けば，まずどうすればいいのか教えてくれるので心強い。

**部下A**「東京ーロサンゼルス間はいかがでしょうか？」

**部下B**「東京ーホノルル間もいいですね」

**部下C**「DC10をもう少し増やしましょう」

**部下A**「もっと整備にお金をかけないと運航に支障が出ます」

山積みになった課題を抱えてうりゃかわ君は会議室をあとにする。

最初からホノルルに3スロット持っているので，まず東京ーホノルル間を開設。3スロットだから週3便だ。続いてロサンゼルスとロンドンに部下を派遣。スロット獲得の交渉開始。機内のサービス教育と整備力強化のために，最大限の投資を行う。結果は……。

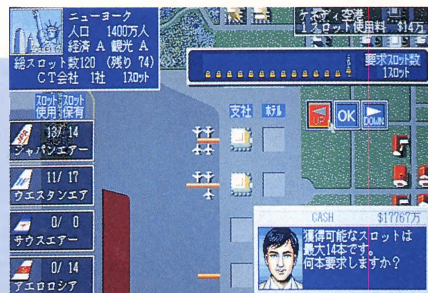
**うりゃ**「うむ，搭乗率は100%。この調子で




X68000用 3.5/5"2HD版2枚組 11,800円(税別)
光栄 ☎045(561)6861




白線が航路。コンピュータのアイコンに注目




他社も乗り入れてくる。負けないように

路線を増やせば大丈夫だな」

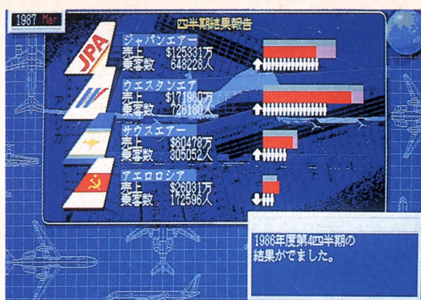
やがて東京－ロサンゼルス，東京－ロンドン線が就航。ロンドンに支社を作って，さらにニューヨークに足を延ばす。

**うりゃ**「うははは，快調快調」

**副社長**「ちょっと経費がかかりすぎてますよ。仕切り率が高くありませんか？」

**うりゃ**「利益は出てるから大丈夫。見ろ，アエロロシアなんか赤字路線だらけだ」

たいていはこういう調子で，順調に売上高を伸ばしていけるのだが……。



売上高の発表。強いぞウエスタン



飛行機は実際のとおり。B747は燃費が悪い

## エアービジネスの真髄

やがて，1984年の夏がやってくる。

**アナウンサー**「ただいま入りましたニュースによりますと，イラクがイランに進攻を開始。両国は戦争状態に突入しました」

**うりゃ**「げ」

石油の値段が急騰。世界の消費は停滞に入る。ジャパンエアーの決算は真っ赤。しかも，同業他社にはしっかり黒字を出してるところがあったりするから悲しい。

**うりゃ**「とほほ，東京－カイロ間の搭乗率が３％しかない」

**副社長**「だからいわんこっちゃない」

たいして需要もないところに週14便も飛ばし，運賃も標準より高くふっかけていたうりゃかわ君に，バブリーな経営のツケが回ってきたのだ。

ここらへんからが「エアーマネジメント」の真髄である。イイ戦争の影響はさして長くは続かないが，それ以上に深刻なのが航空会社の路線同士が競合するようになることだ。お互いの客を奪い合っているうちに，赤字の路線がうじゃうじゃ。必然的に経営の減量化に取り組まざるをえない。

かといって，赤字路線を廃止すればいいというものでもない。客の数を稼がないことには勝利条件は達成できないし，お客さんは航路が多くて，行ける都市の多い航空会社のほうを好む。規模の縮小なんかしてたら，悲惨な末路は目に見えるのだ。

搭乗率が悪かったら，飛行機を小型のものに代えてみる。本数を減らし，浮いたスロットを儲かる路線に回す。また，飛行機もなるべく燃費のいいものに変える。8000km飛べる飛行機なら，3000kmの路線に使うより7000kmの路線に使ったほうが燃費がいい。それでもだめなら，思い切って路線を閉鎖。いろいろ手を入れては，次の決算を祈りながら見つめるわけだ。

**うりゃ**「よっしゃあ。全路線黒字に戻したぜ！」

つらそうな作業だが，これが意外にやっていて楽しい。しだいに効率経営のコツを覚え，自分なりのスタイルが確立されてくる。観光都市にホテルを建てて，その都市へのツアーキャンペーンをやるとか，１ターンだけ戦略価格を敷いて競合他社に手を引かせるとか，いろいろ小技をきかせることもできるようになる。普通にプレイしていれば12年ぐらいで勝利条件達成が見えてくるだろうが，そのときには「今度は違う会社でもっと効率よくやってみたい！」という気になっているだろう。

## 真のビジネスゲームたるために

前作「リーディングカンパニー」でもそうだったが，つまるところビジネスシミュレーションの面白さというのは，「自分の出したコマンドが売り上げや利益につながるところを見る」というところに尽きる。

ということは，コマンドを決めるための判断材料がハッキリしていることと，コマンドと結果の因果関係が明確であることが，ゲームを面白くするための必須条件であるはずだ。さらに，このゲームは現実世界を舞台にしたものだから，どれくらいゲームから現実らしさを感じられるかもポイントになる。

そういう目で見たとき，光栄のビジネスシミュレーションシリーズには不明瞭な部分が多すぎる。ある路線にどれくらいの潜在需要があって，どれくらいが掘り起こされているのか，この都市からはどこに行きたがっている人が多いのか，すべてやってみなければわからない。

政治や経済の情勢がほとんど加味されていないのも不満だ。冷戦の真っ只中なのにニューヨーク－モスクワ線が大ヒットしたりするのは興醒めである。どうも，単純に都市の大きさと人口だけで，需要が決定されているようなのだ。

これがもし，１つひとつの都市の総需要と，ほかの21の都市に対する内訳がしっかり決められていたとしたら，さらにそれが政治情勢や景気の変化によって刻々と変わるとしたら，ゲームのために，思わず図書館に現代史を調べにいってしまうほど，熱中できるモノになっていたのではないかと思う。

現時点でこのゲームが遊べないということではない。ただ，このゲームは比較的高い年齢層を相手にしたジャンルであるはずだ。ビジネスでメシを食っている人相手に本気で熱中させようと思うなら，いまよりもっと高いところに目標を置かなければいけないのではないかといいたいのだ。

ビジネスシミュレーションというジャンルは，日本ではほとんど光栄しか育てていけるソフトハウスがない。歴史シミュレーションシリーズでの立場と違って，冒険も許されると思うからこそ，よりチャレンジングな姿勢でのゲーム作りを望みたい。

### 光栄はゲーム界のマクラーレンか？

このゲームではマウス専用になった。２つの都市をクリックすると，その路線の距離と就航している航空会社を教えてくれる。コマンドツリーが当たり前だった光栄にしては進歩だ。だが，その方法で他社の情報は見られるくせに，自社の路線は見ることができない。調べたければ情報コマンドを使いなさいというわけ。

要するに考え方がなにもかも古い。ゲームそのものを見ても面白いのは認めるけれど，基本的には「信長の野望」のころの発想だ。いまのコンピュータ技術の進歩についていけているとは思えない。

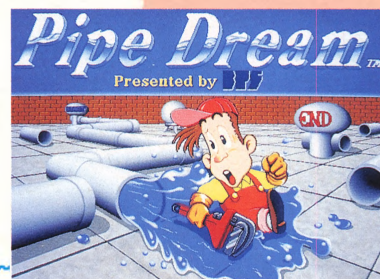
FIでいえば，要するにマクラーレン状態なわけだ。MP4を使い回してここまできて，いい加減限界が見える。もうそろそろなんとかしないと，今日はよくても明日がやばい。

光栄は「信長の野望」の最新作でドラスティックな仕様変更を行ったようだ。僕はそれがシャシーの根本からのフルモデルチェンジであることを期待している。

**総合評価**（0〜10）

| | |
|---|---|
| コンセプト | ★★★★★★★ |
| 操作性 | ★★★★★ |
| スピード | ★★★★★ |
| 熱中度 | ★★★★★★★★★ |
| 新鮮度 | ★★★★ |

# パイプ道を極めろ!




Kageyama Hiroaki
影山 裕昭

**またまた海外からパズルゲームが移植された。数年前にルーカスフィルムが発売した「パイプドリーム」だ。水道管工となった君は，ランダムに現れるさまざまな形のパイプをつなぎ合わせて，水をこぼさないように導くのだ。**

水道管をつなぐゲーム。こういうと皆さんは何を思い浮かべるだろうか。カードの水道管ゲーム，それとも「フリスキートム」のシャワーのおねえちゃん？

この「パイプドリーム」は水道管をつなぐパズルゲームである。ルールは非常に単純。ゲーム開始直後，11×8のプレイフィールドにいくつかの水道管と一緒に，必ずスタートパイプが置かれている。

ゲーム開始から一定時間経過すると，スタートパイプから水が流れ始める。その水をこぼさないように，ランダムに現れるさまざまな形をした水道管をスタートパイプにつないで水の通り道を作っていく。

ステージごとに設定されている最低通過パイプ数だけ水を通すことがステージクリアの条件。1レベルは4ステージあり，後半2ステージは最低通過パイプ数をクリアするだけでなく，プレイフィールド上に置かれているエンドパイプに水を導かなくてはいけない。

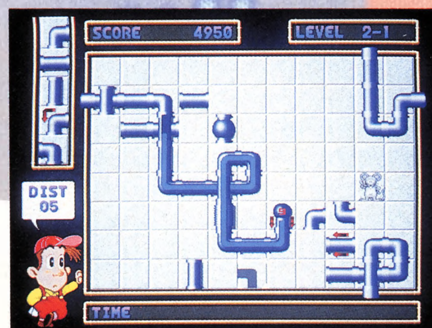
というふうにルールは簡単であるが，やってみると難しい。ランダムに現れるパイプは4つ先のものまで画面左上に表示されているので，そのパイプをどう組み合わせればいいのかを頭で考えてしまうけど，こういうのは直感でやらなければだめ。悩んでいたって一定時間が経過すると無情にも水は流れ込んでくる。そのとき水道管がつながってなくて「あ～あ」と叫んでみてもあとの祭り。クリア条件を満たしていなければゲームオーバーである。画面は真っ青になりプレイフィールドは水に埋もれる。泡が浮かんでくると，それが“GAME OVER”の文字へと変わり，また泡となって消えていく。あ，そうそうユーザーディスクを作成するので，ブランクディスクが1枚必要となっている。



使うパイプによって得点が違う

### もう1回やりたいけど

操作はキーボード，ジョイパッド，マウスと3種類のデバイスからできる。操作性の面では素早い動きを必要とすることから，マウスを使うと具合がいい。全部でいくつのレベルがあるのかはわからないが，途中でゲームオーバーになったとしても，どこまでプレイしたかはユーザーディスクに記録される。次回ゲームをするときは，タイトルメニューで「ゲームの続きをする」を選択する。



水の流れにハラハラドキドキ

ゲームを進めていくなかで面倒なのは，ゲームオーバーになるとスコアベスト10を表示し，タイトル画面に戻ってしまうことである。ゲームオーバーになったステージからリトライすることは誰でも多いと思うが，ディスクアクセスが多く再びプレイフィールドに戻るまで1分ほど時間がかかる。私が下手だからなのだが，これを5回も6回も繰り返したときにはうんざりしてしまう。ゲームオーバーになった時点で「再挑戦しますか？」としたほうが便利だったに違いないと思うと残念である。

さらに細かい部分だが，一度ゲームを始めたらタイトルメニューでも，デフォルト選択が「ゲームの続きをする」になっているとよかった。こういったリトライに対する「遊びやすさ」が徹底されていると遊びやすかったのだが。

ゲームを盛り上げる演出も用意されている。4ステージクリアするごとにマスコットキャラクターによるコーヒーブレイク，プレイフィールドをすべて水の通ったパイプ片で埋めるとボーナスゲームがある。ボーナスゲームはスロットマシンゲームらしいが，残念ながら自分の目で確かめることはできなかった。

X68000用 5″2HD版2枚組 7,800円(税別)
BPS ☎045(472)5815

## パイプ道は険しい

このゲームの面白さは，パイプが完全につながっていなくて水が流れ込んできたときのハラハラドキドキである。これは遊んでみればわかってもらえると思う。ゲームの難易度は「やさしい」「ふつう」「むずかしい」と3段階から選べるようになっている。

不満点は本文中で述べたとおりである。あるステージでハマッて再挑戦するときに，イライラすることが多い。ゲームに対する熱中度も半減してしまい，大きなマイナス要因であることは否めない。

こういったゲームは過去にも発売されたが，遊ぶたびにチクタクバンバンを思い出してしまう。それにしてもスロットマシンで遊びたかったな～。

| 総合評価 | 0 5 10 |
|---|---|
| グラフィック | ★★★★★★★★ |
| 操作性 | ★★★★★★★★★ |
| 興奮度 | ★★★★★★★ |
| 熱中度 | ★★★★★★ |
| コーヒーブレイク | ★★★★★★★ |

# 正統派のパチンコです，ムシュー




あなたの町にもひとつはある，そんなごく身近な遊技場がパチンコ店というもの。「パチンコはやっぱり景品がないと」という人もいるだろうけど，全財産スッてしまったとか，18歳未満だとかいう場合にはコレで勝負だ。

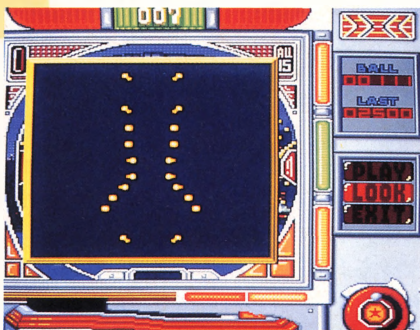
Shibata Atsushi
柴田 淳

皆さんは暇なときに何をするだろうか。僕の場合は本を読むとかビデオを見るとか，たいていそんなふうに他愛なく過ごしているのだが，ごくたまに，暇で，しかも何をするかと考えるのもおっくうなくらい怠惰なときというのがある。

で，そんなときに何気なく楽しめるお手軽なゲームがあるとたいへん重宝である。この「パチンコワールド」は，まさにそんなゲームなのだ。ただしここでいうお手軽というのは，決して軽く作ってあるという意味ではなく，そうアツくもなれないが飽きもしない，というくらいの意味である。

パチンコゲームといえば，マップ上に散りばめられた店舗ごとに既定台数を打ち止めしていき，それにつれてストーリーが展開していくというのが尋常な路線で，このゲームもそこらへんのところを踏襲している。ただ，ストーリーを進めるごとに女の子の服がはだけていくというような，お色気路線のストーリー展開は意識的に避けられていて，正統派の道を突っ走っている。なにやらフランス紳士風に作ったオジサマが突然現れて，主人公の彼女をさらっていってしまう。なんだか「笑ウせえるすまん」みたいな強引な展開だが，なかなかシュールでそそるのだ。

さて，そのストーリーのほうはどちらかというと副次的なものなのだろう。肝心なのはパチンコ台の再現具合である。こちらも目新しいことはあまりないが，しかしよい意味でパチンコゲームらしく仕上がっている。見たところ4種類ほどの台に，また背景や色づかいなどを含めると十数機種分楽しめる。んで，この背景がボカシ気味の取り込み画像を切り張りしていたりして，遠近感があってなかなかいい。



こういう釘の開いた台を探そう



なかなかリアルなパチンコ台

台の釘の具合をルーペでのぞくこともできる。ちなみによく出る台というのは，ハの字型の釘がしまっていて，入り口すぐ上の釘が上向きであるものだそうである。

## パチンコの内部構造に迫る

細かいことだが，このゲームはマウスでの操作ができる。マップ上の移動などには少々使い勝手が悪いけど，玉を打ち出す強さをマウスで操作すると，これがなかなか心地よい。アナログ感覚なのだ。でも，マップ上に落ちているコインを拾ってしまうと，ハンドルを固定できるようになるので，あとはただひたすら玉の行方をボーッと眺めるだけになる。あんまりボーッとしているのも気がとがめるので，僕はこのパチンコ玉の動きはどんなふうにして実現してるのだろうか，などと考えてみた。

パチンコをシミュレートといっても，処理速度にも限界があるわけだから，物理作用を厳密に計算しているのではないはずだ。かといって，そこらへんをあんまりアッサリといなすと，変に現実感をそいでしまう。つまり，厳密性と疑似性を適当なところで折り合わせなければならないのだ。

たとえば斜めに並んでいる釘があるとして，そこに玉が当たると，入射角によって跳ね返る場合と，流れる場合がある。この場合分けがうまくいき，しかも処理が重くならないような方法があればいいのだ。

たぶんパチンコゲームというのは，台の画面のほかに裏マップのようなものを持っているのだろう。で，それというのは玉が流れる方向とか跳ね返る方向などが符号化されている，いってみれば気流のような働きをするようになっているのではないだろうか。こうしておけば，さほど重たくなく，それでいてウソっぽくもなく，モリモリと玉を動かすことができるはずである。

とはいえ僕自身，この方法に確信があるわけではないので，誰か詳しく知っている人がいたら教えてください。頼みます。

X68000用　3.5/5"2HD版　4,800円(税込)
ブラザー工業(TAKERU)　☎052(824)2493

### 武尊の夢

難点をひとついえば，こういうゲームというのは，いったん解いてエンディングを見てしまって，それからは忘れた頃に引きずり出してボーッと遊ぶというのが普通だと思うのだ。しかしこのゲームの場合，その解くまでが長くかかるのである。クリア条件の設定が結構キツめなのだ。苦労させたほうがプレイヤーの達成感が大きくなると考えるのは，少々短絡的であるように思えるが。

このゲーム，パチンコゲームとしての水準には十分達しているとは思うが，その反面個性がない。全体のバランスを崩さない程度に，たとえば玉の動きをものすごく滑らかにするとか，独自のウリがあってもよかったのではないだろうか。

**総合評価**（0　5　10）

| | |
|---|---|
| パチンコらしさ | ★★★★★★★★ |
| 台の背景 | ★★★★★★★★★★ |
| 音楽 | ★★★★★★ |
| お手軽度 | ★★★★★★★ |

# AFTER REVIEW

カプコンのX68000参入第1弾の「ファイナルファイト」。完売のため手に入れられなかった人もいて，再販を望む声も多いようです。期待どおりの熱いゲームに，友達どうしや兄弟姉妹，親子でハマった人も……。

## ファイナルファイト

▶殴って殴って殴りまくる爽快感。
小川　靖浩(21)東京都

▶画面に敵が4人しか出ないのはつらいけど，音楽がシェイプアップされて聴きやすいのでGood。　川名　高司(16)宮城県

▶子供と仲よくやっています。X68000は名実ともにパーソナルゲームステーションと化します。　高橋　茂(42)静岡県

▶敵が吐き捨てたガムで体力回復ってのがシュール。　金丸　勉(20)滋賀県

▶なんといっても2人プレイができることでしょう。あと，「うおお～」とか「ふんふん！」とかのサンプリングもいい。口癖になってます(笑)。川畑　文樹(18)神奈川県

▶メトロシティの町の汚さがよい。
上野　政幸(18)京都府

▶ゲーム音痴の妹が結構ハマっています。やはり操作はシンプルなのが一番ということでしょうか？「ストIIを移植してほしいから」とかいう理由で推薦するバカヤローはエディ・イーのガムでも煎じて飲みなさい。
松本　拓司(18)埼玉県

▶個人的には得意ではないが，X68000を持っててよかったと思わせるものがある。
田沼　基司(26)茨城県

▶やった～！　クリアしたぞ～！　難易度は「EASY」で，ボーナスは10万点ごと，自機(？)は4機，それでもクレジットは使い果たし，残機もゼロだったけど，とにかくクリアしたぞ～！　予想外に感動的なエンディングにはビックリしました。ちなみに，私はボスキャラやアンドレのような体の大きいキャラをやっつけるのも好きですが，ポイズンのきゃしゃな体をふんづかまえて，パイルドライバーで押しつぶすのも好きです(わ，私は変態ぢゃないぞ！)。
敵の体自体に当たり判定がないのは，ドジな私にはとてもありがたい。「パロディウスだ！」じゃ，いつもチチビンタに踏みつぶされたもんな～。それにしても，ジェシカが父親似でなくて本当によかった。
池田　譲太(24)大阪府

▶すごい！　10MHzでも全然スピードダウンするところがないぞ！
樫　正一郎(21)埼玉県

▶ゲームセンターでは聴きとりにくかった音楽がゆっくりと聴け，アーケードと遜色のない仕上がりだ。　住　浩志(18)兵庫県

▶ハガーたちの手によって粛清された街は，これからどうなるんだろう……？
西崎　貴博(17)北海道

▶ハガーのプレスで思いっきり抱きつかれたい。　千貝　健(18)東京都

▶友人と燃えすぎてパットをダメにするくらい楽しめた。　相田　正彦(24)神奈川県

▶1面の地下がアーケードと違ってちらつかないのでやりやすいかな？
甲斐　康彦(21)大分県

▶イカス！　MIDIよりも内蔵音源のほうがよい。　太田　貴道(19)静岡県

▶HD対応がよい。ドライブのどこに入れても絶対起動する賢いシステムが◎。
新井　政樹(20)東京都

▶こんなゲームを作ってみたい。すげーっ！　江崎　諭(19)北海道

▶ハガーの必殺技が，酔っ払ってわけのわからんことをしているオッサンみたいで好き。　北川　悟(16)島根県

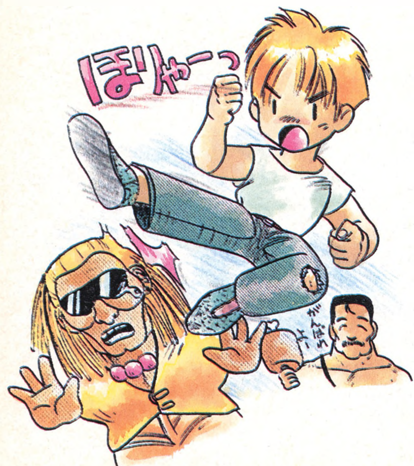
▶ザンギエフのダブルラリアットが見られる。　舟田　宏行(21)北海道

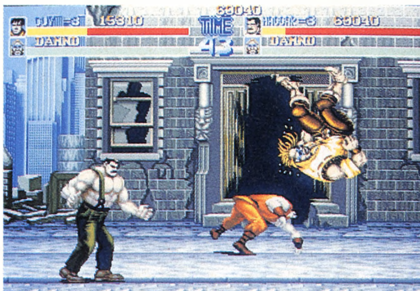
▶手抜きのないところがよい。
福永　浩司(21)大阪府

▶熱い，熱すぎるぜ，X68000が燃え尽きるかも？　今村　雄治(19)岐阜県

▶そーか，エンディングってあなのか(アーケードで見ることができなかったから)。
主藤　二裕(24)福岡県

▶うーん！　殴って蹴って投げて，たまに味方も殴って蹴って(笑)、すげー気持ちい






い！　すでに心は次回作！

本多　登(20)長野県

▶日本車がかわいそう。

高木誠司(17)群馬県

▶２PLAYERでタコ殴り(敵をはさんで連射パンチ)が，思いきり気分がいい。

東　寛(29)愛媛県

▶パイルドライバーがいい。スクリューパイルもしてみたいぞ，カプコンさん。

新子　弘康(21)大阪府

▶３人いるし，エレベーターは動くし。

林　順一(18)千葉県

▶熱いゲームだ。連射スティックのない私は本当に汗をかくぐらい熱くなる。

浅井　保博(22)神奈川県

▶100円玉もしくは50円玉の数を気にしないでやることができる！

大野　豪隆(19)千葉県

▶ケンカもののゲームが欲しかったので，つい買ってしまった。クリアしても飽きないのがいい。　中垣　敦(18)兵庫県

▶兄妹でやると対戦になってしまう。しかもコンティニューがなくなる。

松永　好司(17)富山県

▶破壊的な面白さ。プレイしながらなぜかゲラゲラ笑ってしまう。

吉本　康孝(20)福岡県

▶市長(ハガー)はどう見ても市長らしくない。　姉帯　寛(23)神奈川県

▶どーせたいしたことないと友人宅で遊んだら，夜が明けてしまった。シクシク。

西方　茂樹(23)茨城県

▶ハガーのフライングボディアタックがよい。　谷岡　宙(28)大阪府

▶２人同時にできるゲームはいい(特に協力型)。　仁井内　明(23)広島県

▶エンディングがすごい。「タッチ」(ＴＶのほう)の最終回のラストシーンを思い出してしまった。　松永　正弘(22)京都府

▶スーパーファミコン版になかった４面，２人同時プレイ，階段の下りなどのところもちゃんとあってよかった。敵を殺してストレス解消！　が，音楽はだめ。

横山　崇男(15)埼玉県

▶エンディングで，ガイがコーディーを殴る蹴るする気持ちがわかる。

宇賀　崇人(21)大分県

▶面白すぎる。最初は最終面クリアが目的だったのでコーディーを使っていたが，最近ではハガーがお気に入りです。コンティニューなしでどこまでいけるかにチャレンジしています。巷では次はSFIIだといっていますが，私はFF II(仮題)も捨て難いと思います。とにかくアフターバーナー以来のハマりようです。鎌田　俊之(27)埼玉県

▶ゲーセンに行けば50円でできるのに，と思われがちだが，接客にはもってこいだろう。　高野　真樹(20)山梨県

▶やり込めばやり込むほど，プレイが美しくかっこよくなっていく。

美辺　央希(18)東京都

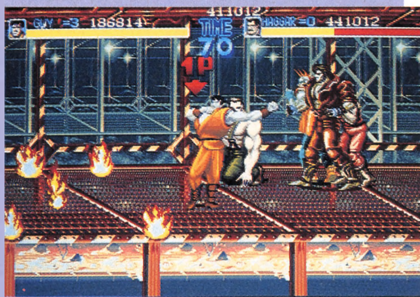
▶最初は教育上よくないと思い，買うのにも抵抗があったが，一度プレイしたらハマってしまった。平和主義のあなたにもオススメ！　中村　康彦(35)石川県










## 発売中のソフト

★エアーマネジメント　光栄
X68000用　3.5/5″2HD版　11,800円(税別)

★オーバーテイク　ズーム
X68000用　5″2HD版　9,800円(税別)

★ロードス島戦記II　ハミングバードソフト
X68000用　5″2HD版　9,800円(税別)

★チェイスH.Q.　ブラザー工業(TAKERU)
X68000用　3.5/5″2HD版　7,800円(税込)

★ストライダー飛龍　カプコン
X68000用　5″2HD版　9,800円(税別)

★スクエア・リゾート　ハイパー戦車戦　ファミリーソフト
X68000用　5″2HD版　4,500円(税別)

★究極タイガー　KANEKO
X68000用　5″2HD版　8,800円(税別)

★キングス・ダンジョン　ソフトプラン
X68000用　5″2HD版　5,800円(税別)

★パイプドリーム　BPS
X68000用　5″2HD版　7,800円(税別)

★パチンコワールド　ブラザー工業(TAKERU)
X68000用　5″2HD版　4,800円(税込)

## 新作情報

★エトワールプリンセス　エグザクト
X68000用　5″2HD版　9,800円(税別)

★Traum　M.N.M Software
X68000用　5″2HD版　価格未定

★沈黙の艦隊　ジー・エー・エム
X68000用　3.5/5″2HD版　12,800円(税別)

★餓狼伝説　ホームデータ
X68000用　5″2HD版　8,500円(税別)

★ロボスポーツ　イマジニア
X68000用　5″2HD版　価格未定

★シムアント　イマジニア
X68000用　5″2HD版　価格未定

★ヴェルスナーグ戦乱　ファミリーソフト
X68000用　3.5/5″2HD版　9,800円(税別)

★ドラゴンスレイヤー英雄伝説　SPS
X68000用　5″2HD版　価格未定

★鮫！　鮫！　鮫！　KANEKO
X68000用　5″2HD版　価格未定

★達人　KANEKO
X68000用　5″2HD版　価格未定

★エアバスター　KANEKO
X68000用　5″2HD版　価格未定

★サバッシュII　ポプコムソフト/グローディア
X68000用　5″2HD版　価格未定

★極　ログ
X68000用　5″2HD版　12,800円

★メガロマニア　イマジニア
X68000用　5″2HD版　価格未定

★倉庫番リベンジ/ユーザー逆襲編　シンキングラビット
X68000用　5″2HD版　6,800円(税別)

新製品紹介

# SOUND SX-68K

Taki Yasushi 瀧　康史

X68000のFM音源音色作成ツールには「SOUND PRO-68K」があったが，そのSX-WINDOW対応版といえる「SOUND SX-68K」が発売された。ほかのソフトで演奏中の音色をリアルタイムに変更したりもできる。

このソフトのレビューをやることになって，さっそく起動してみようとディスクを入れると，HUMAN.SYSが入っていないとのこと。それなら，とハードディスクから起動してみる。マニュアルにはOPMDRV3を常駐せよと書いてあるけど，昔書かれた欠点は直したとのことなのでZ-MUSIC上のシステムで，試しに実行……。あ，髑髏さんこんにちは。ああ，悲しい。

OPMDRV3に入れ替えて再挑戦。さて，もう一度“SOUND_SX”をクリック。あれ，なぜ？　どうやらOPMDRV3が常駐に失敗したらしい。

念のためにコマンドからOPMDRV3を起動してみると，PCMDRV.Xが常駐しているとのこと。PCMDRV.Xなんていつのまにできたんだろ。まあ，PCMDRV.SYSのことだろうから，これも削除。SX-WINDOWに戻って，もう一度クリック。あ，やっと起動できた。

さしあたっていちばんうれしかったのは，アイコンの登録を自動的にやってくれるユーティリティが付属していたことかな。

アイコンメンテなるものがあると，拡張子によって“.LZH”のついたファイルは「リボンのついた包み」のアイコンとか，“.PIC”ファイルはカラフルな絵のアイコンだとか，X68000バッチを金色にしてみたりとか。まあ，いろいろとやってしまうわけ。

こういうデータは，icon.lbという名前のシステムファイルに保存されるんだけど，人によってアイコンを定義している数や，まだ空いているアイコン番号の値は違う。そこで新しいアイコンが増えてしまうと，パターンエディタなるものを使って手作業で転送し，アイコンメンテでちゃんとそれなりに書き換えなければならない。

つまり，そういうことを自動的にやってくれるユーティリティがあるということ。厳しくいってしまえばこういうのを設けるのは当たり前かもしれないけど，ちゃんとユーザーというものを認めてくれたようでうれしかったな。

これらのことは，これからシャープから発売されるSX-WINDOW対応アプリケーションの指針を，暗黙のうちに示してるような気がしない？

## 節約のMINI

実行ファイルは2つ。当然，ひとつはノーマルのSOUND_SXですが，もうひとつ，機能を最小限にしたミニマムバージョンがついています。これはほかのアプリを同時に使用していたりして，メモリに十分な余裕がないときにも立ち上がるようにサイズが小さくなっているのです。ところがあまりにも最低限の機能しかないため，結局使えないのが現実です。

なんといっても音色エディタと，キーボードしかないんですから。ウリのひとつ，サウンド.Xで再生中の曲の音色をリアルタイムでエディットする，という機能を使っても，結局作成したデータはセーブできないため，紙に書き移さねばならないという原始的な作業に移らなくてはいけません。もちろん，完全機能版を使えばそれですむところですが，せっかくのMINIなのですから，これくらいの機能はつけてほしかったところですね。

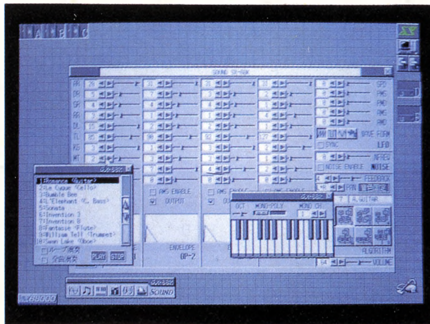
## ウィンドウがいっぱい

MINIの話はこれくらいにして，実際の使い勝手を紹介していきましょうか。

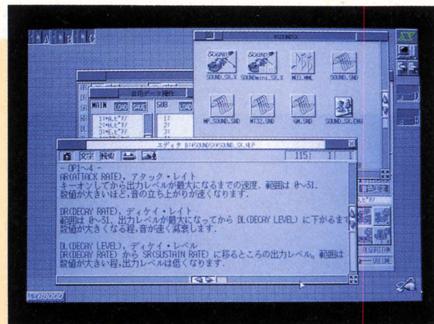
メインウィンドウとなる音色エディタを残して，あとはすべてサブウィンドウです。「Easy Paint SX-68K」と同じく，メニューのサブウィンドウがひとつあって，このウィンドウからプリンタ周りやディスク周りの操作，「SOUND PRO-68K」からの音色のイメージエディット，実際に出力されている波形を表示するウィンドウなどを呼び出すことができます。ちなみに，これらはみんなサブウィンドウなので，音色エディタがいちばん前にいないときはきれいさっ




X68000用　3.5&5″2HD版　15,800円(税別)
シャープ　☎03(3260)1161



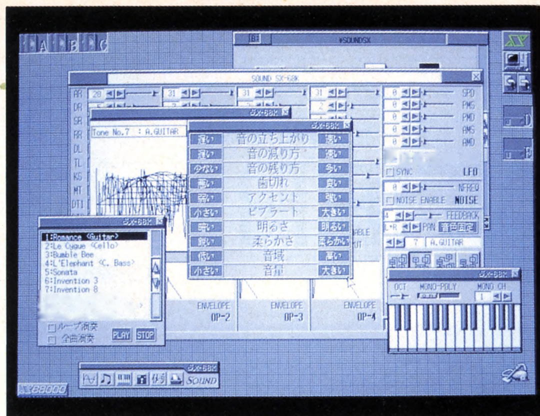
メインのほかにいくつかのサブウィンドウがある



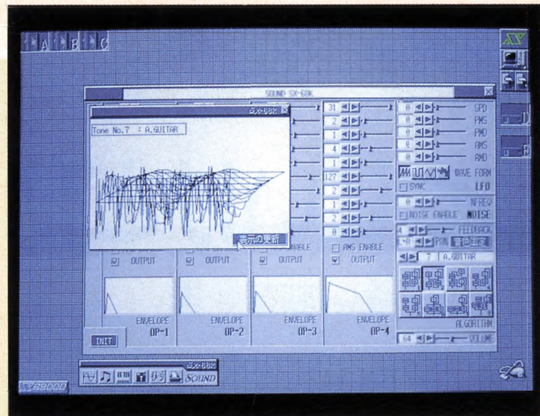
アイコンを自動的に設定してくれる

ぱり消えてしまいます。

前述したほとんどのサブウィンドウを開いていて，ちょっとエディタなどでMMLを書こうと，エディタをアクティブにすると，音色ウィンドウを残して，全部を消す処理に移ってしまいます。うざったい感じがするのは決して私だけではないでしょう。



音色のイメージを言葉で表現するのは難しい



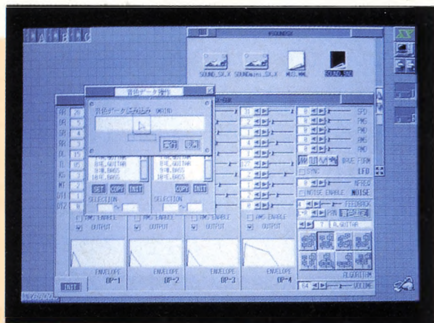
波形表示はやはりかっこいい

SX-WINDOWで走ることや，「適当にやれば使える」という点ではたしかに合格ラインです。「SOUND PRO-68K」とはデータ互換になっていますし，あれからは使い勝手がかなり上がっています。たとえば，波形表示もイメージエディットも，操作しながら別ウィンドウでちゃんと音色エディタのパラメータが見えています。よって，イメージエディットでは，どの項目を変化させるとどのパラメータが変化するのかが一目瞭然なのです。

音色のイメージを言葉で表現するのは難しいことですが，このイメージエディットでもそれに漏れず，何をいいたいのかわからない形容がいくつかあります。「音の減り方」と「音の残り方」は言葉で聞くと似たようなものに感じてしまうし，「アクセント」といわれても，何をどうアクセントするのかを想像するのはなかなか難しいことでしょう。

まあ，いじりながら「なるほどこうするとあのパラメータが動くのか」というように覚えていけばいいのでしょう。しかし，音の柔らかさのパラメータをいじると，音のアルゴリズムが変わるのは，ちょっと的外れのような気がするのですが。

音色エディタのほうは「SOUND PRO-68K」と同じで，音のエンベロープが図で表されています。しかし，気のせいかもしれませんが，実際とはなんだか違うような感じがするのです。それに，エンベロープを目的の形状にするのが結構たいへんで，なかなか思ったとおりにはならないので，新設された小さい横スクロールバーが，だんだん恨めしくなってきてしまいます。



SX-WINDOWだからこんなことも

せっかくマウスを使うのですから，頂点のあたりをドラッグしたりして形を変えると，自動的にその形に合ったパラメータに変更してくれるとかいった操作まで実現されれば，もっと簡単にイメージどおりの音を作れるツールになったと思います。

ちなみに「SOUND PRO-68K」よりもかなり多いサンプル曲（といっても音色のテスト用）がついてくるので，そのあたりはなかなか評価すべきところでしょう。

ただし，ヘルプファイルの呼び出しには問題があります。エディタ.Xで表示されるのですが，ファイル名を与えるための文字列のバッファの制限がたった30文字程度しかないらしく，たとえば“A:￥アプリケーション￥SOUND_SX”というディレクトリに入れてあっただけでも，SOUND_SX.HLPは呼べなくなってしまうのです。もちろん，ディレクトリ名を短くしたり，ルートディレクトリのすぐ下にディレクトリを置けばそれですんでしまうのですが，ひとつのアプリケーションにディレクトリの構造まで制限されてしまっては，独自の環境の「美学」に反します……と，これは冗談ですが，この30文字程度というのは，あまりに少ないのではないでしょうか。

## OPMDRV3.Xとは?

冒頭で述べたとおり，「SOUND SX-68K」にはOPMDRV3.Xという音源ドライバがドキュメントとともに同梱されています。「SOUND SX-68K」を使うには，このソフトを常駐しておかなくてはなりません。

新しく増えたMMLはほとんど，MUSICDRV.XやOPMDRV2.Xのものと同じようなので，多少いじればZ-MUSICとの間でもなんとかデータ交換できるでしょう。

シャープからのアプリケーションにしてはめずらしく，キャラクタデバイス(OPMとMIDI)を登録するにもかかわらず，常駐解除もできてしまうようになりました。それなら，Z-MUSICととっかえひっかえできると思ったらちょっと甘かった。

なんと，OPMDRV3.Xは何かを残しているのか，Z-MUSIC側の問題か，常駐解除後にZ-MUSICを常駐しようとすると，MUSIC BIOSがすでに常駐しているとか，なんとかいわれて怒られてしまいます。

結局，Z-MUSICの環境と「SOUND SX-68K」の環境はまったくの別世界になってしまいました。今後シャープから出される音楽関係のアプリケーションはみんなOPMDRV3.Xの道を進むことでしょう。どちらを選ぶかはユーザーの決めることですが，同じマシン上でアプリケーションをとっかえひっかえできないのは悲しいことです。

### どんどん出して

厳しいことをいろいろ書いてしまいましたが，うまく使えば面白いことができるツールです。SX-WINDOWのツールは3本目で，まだまだ始まったばかりです。SX-WINDOWも，土台そのものはMS-WINDOWSをはるかにうわまわる素質をもっていると私は信じていますので，これからいろいろなツールを出して普及に励んでほしいところですね。

ユーザーが増えるということは，ソフトもそれにつれて発売されやすくなるということですから，いつかおじさんたちが「WINDOWS？ だめだよ，SX-WINDOWじゃなくちゃ」といってくれるような世の中になったらいいな。う〜ん。

**総合評価**

| 項目 | 評価 (0 5 10) |
|---|---|
| 使い勝手 | ★★★★★★★★★ |
| 音色の作りやすさ | ★★★★★ |
| 美しさ | ★★★★★★★ |
| サブウィンドウ | ★★ |

# しつこいけどMATIER

Ogikubo Kei 荻窪 圭

**ここのところ，ほうぼうに立体視の話が出たりして，読者の反響も多いようですが，今月もその話題から始まります。あ，後半は読者投稿の写真を使わせていただいて，加工したりして遊んでいます。感謝。**

週刊文春を読んでいたら，いや，眺めていたら，日産がランダムドットの裸眼立体視を使った広告を出していた。単純なやつで，「L」って文字が浮き出るだけなんだけど，さすが，めざとい。

しかし，さすが丹先生ですね。先月のあのランダムドット立体視プログラム。ひさびさにマジにプログラムを読もうと思ってしまった（思っただけだけど）。

ランダムドットといえば，関係書が出たようである。ワニブックスの「マジックアイ」ってヤツなんだけど，ランダムドット立体視の絵を集めただけ，っていう娯楽ものだったからちょっと残念。面白い本ではあるのだけれど，ノリがね。ランダムドットを立体視する，ってことだけに主眼が置かれている。「うまく見えましたか？」「今度は何が見えますか？」「立体視すればクイズの答えがわかります」ってなもんで，確かに最初は，何もないところから図形が浮き出てくる，っていうことに驚きを感じるんだけど，それをそのままで引っ張っていって，しまいには「今度のは難しいけれども頑張ってみましょう」はないだろう，と思う。ランダムドットの面白さは，予想を超えたものが出現する面白さであり，「わあ，立体に見えたあ」ってレベルで止まっていてはつまんないのである。

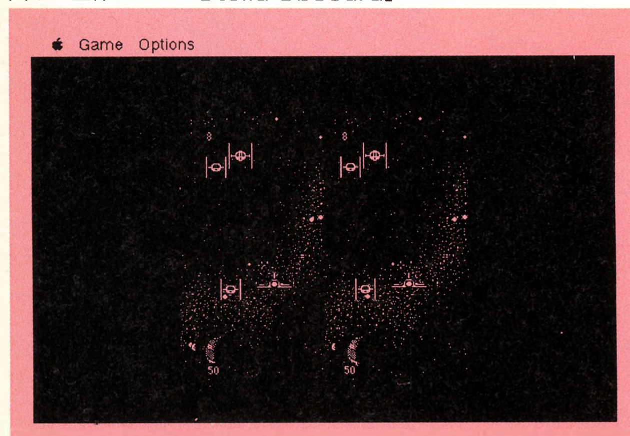
気を取り直して，小学館からもCGによる立体視の本が出ていると聞いて本屋へ行く。見つからない。泣く泣く帰る冬の午後。

## でもって，また立体視の話だ

あれからいろいろと調べてみたりすると，あるわあるわ，裸眼立体視のネタ。ステレオオタク学会なんてものまであって，大阪ステレオ博なんてのも開かれたらしい。すみません，私の勉強不足でした。浅学荻窪圭と呼んでください。いろいろノウハウもたまっているようで，ステレオ写真歴ン年って人も大勢いて，グラフィックソフトでちょこちょこと遊べばなんとかなるだろうと考えた私は甘かった。

最近見つけた最高の玩具が図1の3Dゲーム。見ればわかるとおり，ステレオ映像を使った立体に見える3Dシューティングゲームなわけやね。「Stella Obscura」っていう1990年に作られたソフトなんだけど，これがけっこうバカバカしくていい。ちなみにMacintosh用ね。「Nautilus」という月刊のCD-ROMマガジン（毎月，海の向こうから送られてくるのだ。定期購読しているから）に収録されていたのだ。なんでいまごろ，って思うけど，タイミングよく手に入ってラッキーという感じである。画面はモノクロなんだけど，それがかえっていい味を出していて，敵が迫ってくるのを立体視して迎え撃つ，なんて頭がクラクラする。疲れるんだ，これが。

図1　立体ゲーム「Stella Obscura」




このゲームのドキュメントにはステレオビュワの作り方がついていたのだけれど，やはり，ビュワなど使わずに根性で視線を制御して立体視するのが楽しい。

そうそう，SIONIIのステレオ化なんて面白いと思うのだが，誰かやりません？

そういえば，ホロフォニックサウンドっていうのもあって，こちらは3Dサウンドなんだけど，こいつもなかなか面白い。左右の広がりだけではなくて上下も認識できるサウンドシステムで，先日，「脳にドリルで穴を開ける」なんてのを聴いたもんだから，もう気分は「ぎゃぁぁぁぁぁ」である。怖い怖い。「ウソだ」ってわかっていても怖い。

立体視もホロフォニックサウンドも「リアル」を超えるところがいたく私らの心を刺激してくれるんだよね。ハイパーリアルっていうの？　知らないけど，「ウソだ」ってわかっているんだけど，脳の中にそれを信じちゃっている単純な部分があって，「理性」と「感覚」のせめぎあい，っていうか，「感覚器官」から入っている情報の矛盾，っていうかそのギャップに生まれる刺激ってのが恐怖なのかもしれない。「感覚器官ごとの情報に矛盾がある」って怖いよね。

耳からの情報は「左に声がする」で，目からの情報は「右に人がいるけど左は崖だ」だったりして，行動の記憶は「ここには誰もいない」であり，筋肉からの情報は「目は遠くを見つめている」だったりして，そういった情報が脳の中で矛盾して，どれかに身を委ねられればいいのだけれど，その矛盾が相克したまま情報は入り続ける。

バーチャルリアリティって，そういう恐怖を人に与えるような気がする。いまはまだ他愛ないからいいし，ゲームなんかだと「リアルな仮想空間」って楽しいレベルで遊

んでいるけど，その気になったら人格の崩壊くらい簡単にできてしまうような，そんな可能性もあると思うのだ。

まあ，さまざまな外界からの情報の矛盾が気にならなくなったときが，いちばん怖いのだろうな。

## 平行法と交差法

話は立体視に戻るのだけれども，ヘンなグラフィックソフトを見つけた。マウスで絵を描くのだが，それが，自動的に裸眼立体視用ステレオ画像を作ってくれるのだ。左と右の枠があって，どっちかに絵を描くと，反対側にちょっとずれた絵が自動的に作られるの。これは，マウスを動かすスピードがそのまま奥行きになる，って寸法だ。時間をZ方向の距離に置き換えるパラダイム変換だそうで，難しいことはようわからんがなかなか面白いもの。こいつもMacintosh用。

誰でも考えそうなネタではあるが，面白い。「役に立たない」っていいよな。

さてさて，立体視の話は一度したのだけれど，もう一度，補足説明。

裸眼立体視，つまりステレオ写真には2種類ある。平行法と交差法である。って書くのは3回目のような気もする，が，まあ，いいとしよう。

私が平行法しかできないので平行法の話しかしなかったわけだが，やはりこのあたりはちゃんと押さえておかねば，と思い直したのだ。

平行法はステレオ写真などでお馴染みだが，大いなる欠点を持っている。人間の目と目の幅である。人間の左目と右目の間隔というのはだいたい6cmから6.5cmくらいなのだ。人によってそれぞれとはいえ，10cmの人とか3cmの人，ってのはたぶんいない。平行法というのは左目で左の画像を，右目で右の画像を見るものだから，もし，左右の画像が何10cmも離れていたら……こりゃあたいへんである。

人間の目っていうのは，近くのものを見るときには寄り目をするが，遠くのものを見るからといって離れ目(っていうのか？)になるわけではない。どんなに遠くを見るにしろ，視線はせいぜい平行線であり，広がっていく必要はないのだ。だから，ふだん使わないような筋使いを要求されるためにたいへんなのである。だいたい，左の画像と右の画像の対応する点が6cm以上離れると，平行法は難しい，といわれる。

ってことは，大きな絵だと平行法では難しい，ってことなのだ。大きな絵でも遠く離れて見れば視線は平行線へ近づいていくから楽にはなるが，何メートルも先の絵なんて近眼の私には見えない。

パソコンのモニタってのはX68000の場合，おおむね15インチであるから，平行法用の立体視画像を作ろうと思うと，けっこう小さなものになってしまわざるをえない。だから，今回の図も小さいのね。フルスクリーンで平行法用の画像を作ってみても，だれも立体に見ることなんてできない。

ランダムドットの場合は，左目用の画像も右目用の画像も1枚の絵の中に組み込んでしまうから，でかいものが作れるのだ。

その点，交差法はでかい絵に有利である。なにせ,「画像と目の間に焦点がくるように視線を合わせつつも，ピントは焦点の向こうにある画像に合わせる」などという超絶技巧（と私は信じているが）を使えばいいだけなのだ。視線はクロスしてどんどん広がっていくため，その人の寄り目能力にもよるが，かなり大きな画像でも立体視できるというメリットがある。

交差法の訓練として，知り合いのイラストレータは画像と目の間にエンピツか何かを置いてそれをじっと見つめ続ける，っていっていた。とにかく，画像は無視して，エンピツを何分も見続ける。そして，目と脳に視線を染み込ませるのだ。せっかちな人（私のことだ）には無理だ。

しかし，交差法から入った人は，平行法は難しい，という。うーん，人間って不思議である。

大手のネットには数多くのステレオ画像データがアップされているが，パソコンの画面は平行法で見るにはでっかい，ってこともあってか，交差法も多い。考え方しだいだね。交差法のほうが大きな絵を見られるから，けっこう粗いパソコンの画面に向いているといういい方もできるし，平行法は大きさに制限があるから，画像ファイルが大きくならずにすんで通信には向いている，っていい方もできる。

ステレオ画像の面白さは，その非現実感にあると思っている。3D化によって非常にリアルな像が脳内に結ばれるわけなのだが，そこに漂う妙な胡散(うさん)臭さが面白いのだ。ランダムドットなどはまさにそれで，ランダムドットの空間だけが「立体」である，というのはホログラムに似ている。窓の中だけが立体，という異世界が面白いのだ。

## 今月のMATIER

などと書くと「また立体視の話？」って思う人も多いだろうが，今回は1枚だけ。せっかく「東京都中野区の江畑」さんが送

### CG STEREOGRAM




本文でちょっと触れようとして失敗した「CG STEREOGRAM」（小学館）を買ってきた。間に合わなかったのでコラムになってしまったわけやね。これが「マジックアイ」など比べものにならないほど面白くて難解だ。とにかく，難解である。著者紹介のページに収録作品の難易度について書いてあったけど，本全体の難易度は高いと思う。高い。でも，面白い。

簡単に見えるけれど中身がつまんないステレオグラムは暇潰しにもならないけれど，この本は暇潰しどころか，人生を変えかねない衝撃を与えてくれる。より無駄な人生を送るには最適な本だ。

本書はパラパラとめくるのが正しい。最初から難解だからだ。カラーフィールドステレオグラムである。ランダムドットに似ているけれど，違うテクだそうだ。なにしろ，カラーである。続いて，ランダムドット。なんとPC-9801のBASICとMacintoshのHyperTalkのプログラム付きという親切さで，いろいろと説明してくれる。続いてウォールペーパーステレオグラム，ステレオ写真などを経て，日常の中からステレオグラムを発見するという傑作を経て，最後にマスター講座という構成。

まあとにかく，頭は痛くなるわ道を歩いていて突然眼が立体視状態になってしまうわという副作用はあるにしろ，成功するまでは眼の制御の難しさや脳の不思議議を，成功すればしたで新しい楽しみを発見できるというわけで，とりあえず，買うべきである。　1,800円(税込)

ってきてくれた水着PICファイルであるからして，使わなければ，ってわけなのだ。江畑さん，どうもありがとう。

私は宣言する。「大人のためのX68000」は「Oh!Xのスーパー写真塾だ！」というわけで，どんどん投稿をお待ちしております。アブないものでもOKです。誌面に載せると警察がとんできて編集長がこってり絞られるようなものもOKです。そういうのは掲載しないで，私が隠匿してひとりで楽しむから。えへへ。

そういうのは「大人のためのX68000 スーパー写真塾化計画係」まで。「大人のためのX68000 セーラーメイトDX化計画係」でも「大人のためのX68000台風クラブ化計画係」でもいいぞ。

さてさて，いろいろと写真やらPICファイルやらが送られてきたのだが，サングラスをかけてバスタオルをはおっているヤツをとりあえず採用した。

元の写真からタオルをはおった女の子だけを抽出し，縮小し，コピーして立体化する。ってのが手順。それでもって，背景用の別の絵と重ねるわけだ（今回はセンチュリーハイアットホテルね）。

まず，元の写真から女の子を抽出する。これにはMATIERの「スムーズ閉曲線機能」が非常に役立つ。ベジエ曲線で閉曲線を指定できるのだ。こいつはすこぶる便利。ベジエだから，適当にポイントしておいて，あとで曲線ポイントを細かく修正できたりして，遊べる。欲をいいだすときりはないが，まあ，これなら使えるだろう。

ルーペで2倍にし，ベジエ曲線を使って輪郭を切り取る。これは，フォトレタッチソフトで有名なMacintoshのPhotoShopでもver.2.0でやっとサポートされたほどの機能だ。

で，背景を適当な単色で塗って，ルーペを解除する。このとき，元の倍率へ戻るときに自動的にアンチエイリアスがかかる。なかなか楽しい。

背景を塗り終わったら，大きさを調節して，コピーして，メッシュ変形をしたり，コピーを使ったりして，立体に見えるっぽくする。実際に見えるかどうかは別だぞ。


女の子だけを取り出して立体化させる

部分的に浮かび上がらせるっていうのはけっこう難しいのだ。なにせ脳味噌の経験値がそういうデータを受け付けにくいもので，全体がちゃんと立体映像になっていればいいけど，それもなかなかうまくいかない。

それが終了したら背景にマスクをかける，のだが，「アンチエイリアス」のおかげで，背景と前景の間にヘンな線が入ってしまう。境界線が調整されてしまうからね。これは，マスク膨張などを使ってごまかす。で，マスクを反転させる，のである。いっちょ上がり。で，裏画面にコピーしてやる。

続いて，表画面には背景用のデータを読み出す。大きさや位置を合わせながら，立体用背景にする。

裏画面と表画面を入れ替えると，マスクされた女の子が表画面にくる。

表画面に裏画面をコピーする。すると，マスクされていないところに裏画面が貼り付けられる。マスクをはがす。

これで終わりである。なんて簡単。やっぱ，メモリをガシガシ搭載して，裏画面を4枚使いまくればいいのね。こいつは楽チンである。

しかしまあ，うまく立体には見えないものでたいへんだ。もともと脳味噌をだまくらかして（つまり，脳の長年の学習を悪用して）見ているようなものだから，見えにくいものは確かに存在する。

これ，ディスプレイだとあまり立体に見えないんだよね。印刷だとどうだろうか。

## せっかくの水着なので

あけましておめでとうございます。今月は沖縄から生中継です。

んなわけないが，水着である。このタオルをはおっている女の子。だいたい，泳ぎにいくのにニコノス持っていくヤツ。わざわざ水に潜って撮るヤツ。水ん中でピースするヤツ。いいんだけど。

それでちょっと遊んでみた。

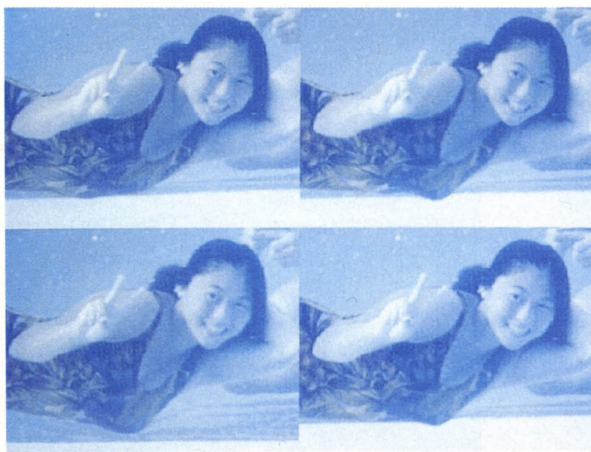
それが，水中4分割写真。

いやね，その女の子（名前はなんていうんだっけ。え，エツコ？）が水の中でちょっと反っているものだから，胸なんかがあまりきれいに出ていない，ってんで，ちょっとメッシュ変形させてもらいました。

どれが元の絵でしょう。

胸を大きくしてみたのはいいけれど，そうしたら，乳首が床にめりこんだみたいになっちゃって，あまりに不気味なので，左下のヤツはちょいと床面を下げた。


そこにマスクを使って背景を合成する


メッシュ変形機能で水中女性4変化

これはほんの暇潰し。

続いて，最初のタオルはおりの写真。せっかく背景から女の子だけ切り取ったのだからってんで，ほかの絵と重ねてみた。

まずは派手なやつ。

下半分は「誰でも描ける秋の情景」ってやつね。スプラッタブラシを少し秋の草原っぽい色，長さにして，自動的に描かせる。それだけ。適当。んでもって，やめる。

空は青。太陽は球。あまりにヘンだから，空に3Dペインティングを施してみた。もっとヘンになった。空にめりこむ太陽，ってんで。でもって，ちょいとばかりオブジェを浮かべる。

3Dペインティングって，面白い。上にタオルをなびかせた女の子をかぶせる。

お次。裏画面から表画面にコピーするとき，あるいは同一画面でも任意の範囲をほかの場所にコピーするとき，透明度の指定ができる。これとクリップコピーを組み合わせれば，いろいろと遊べそうだ。

クリップコピーってのは指定した色を透明色としてコピーする機能で，黒か白を透明色に指定することができる。

で，下半分に，せっかく送ってきてもらったPICファイルをいろいろと重ねてみた。印刷できれいに出ているかどうかはわからないけど，まあ，習作ってことでお許しください。

ついでに，女の子も70％くらい（だったかな，忘れた）で左につけてみた。右のほうはちょっと遊んで，乳首あたりに先々月に使った私の眼を埋め込ませてもらった。そんだけ。

この，半透明の絵を重ねていく，ってのは遊べそうだな，と思う。

## 細かい心配りは誰のため？

MATIERの面白いところって，細かい気配りだと思う。愛があるのだ。

たとえば，暴走する。まあ，時々暴走する。起動し直す。で，そのときの画面は残っているわけだけれども，裏画面やアンドゥ画面は残っていない。しかし，メモリがクリアされていないかぎり，あるいは，暴走時に画像データの入っていたアドレスに悪さしていないかぎり，論理的にはデータは残っているはずだ，ってんで，GRESTOREってプログラムがついてくる。MATIERは実はテンポラリファイルに裏画面やアンドゥ画面で使ったメモリの先頭アドレスを記録してくれるのだ。で，GRESTOREを使えば，そのアドレスをもとに復活してくれるわけ。このへんの発想がよいよね。

さらなる愛としては，JPEGのサポートがある。パスを切ったディレクトリにフリーウェアとして配布されているJPEGのローダ/セーバがあれば，MATIER上からそれを使ってJPEGファイルのロード/セーブができるのだ。小さな親切，うれしいお世話である。これで，MacintoshやPC-9801のフルカラーボードで取り込んで作られた写真データも，ダウンして遊べるってもんだ。

さらにさらに，Rキーを押すと，768×512ドットで65536色だけど，実際にアクセスできるのは512×512ドット分だけだよモードに切り替わるのである。これやると，ドットの縦横比が1：1になる。つまり，640×480ドットのパソコン（MacintoshやPC/AT，FM TOWNSなど）で作ったグラフィックデータを，元の形のままで見て，加工することができるのだ。さすがだね。

裏画面ペンモードってのも遊べる。裏画面が任意のペンで表画面へコピーされる。

ほかにもスクリーンセーバーがついているとか，マニュアルにフジカラーフィルム出力サービスの使い方が載っているとか，細かい気配りがあってうれしい。スクリーンセイバーはfunsuiが私の好みである。ま，それはどーでもいいか。

結局，年賀状作らなかったわけだけど，まあ，人生，そういうものやね。でも，投稿年賀状にはきっとあるだろうなあ。MATIERでCMY分解出力してプリントごっこしたやつとか，フィルム出力サービスを通してそのまま写真年賀状にしてしまうやつとか。まあ，時代はまだまだ面白くなる，ってことで，さようなら。


背景に変な絵を合成して遊んでみた


透明度を調節して重ね合わせた背景



アスペクト比が１：１になるモードは便利

新製品紹介

# POLYPHONとはなにか?

Kageyama Hiroaki 影山 裕昭

POLYPHONは本体のCPUと並列して動作するサブCPUボードです。拡張メモリ，数値演算プロセッサ，MIDI出力，PCMダイレクトアウトなどの機能を1枚のボードに凝縮したコストパフォーマンスの高さも魅力です。

スロット不足に悩む人は多いだろう。

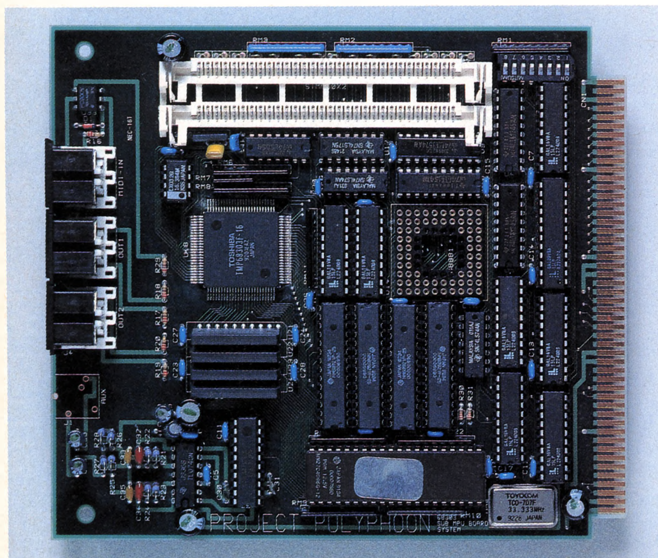
なにをおいてもメモリは必要だし，DōGA CGAシステムでアニメでも作ろうかということになれば数値演算プロセッサがあったほうがいいし，やっぱりMIDIもほしい，ついでにSCSIも……とやっていくととてもじゃないがスロットが足りない。1枚のボードの上にたくさんの機能を詰め込んだボードはできないだろうかと考えた人もいたのではないだろうか。

今回紹介するPOLYPHONはそんな集合合体ボードとしての側面も持っている。

ボード上に2Mバイトないしは8Mバイトの増設RAMが載っており，数値演算プロセッサ用ソケットも標準でついている。数値演算プロセッサは秋葉原なら2万円で十分おつりがくる値段である。

さらに，8Mバイト版は単純にRAMボードとして考えても破格に安い。単純にRAMボードとして考えた場合，現在2Mバイト拡張RAMがツクモ特価33,800円（本誌広告参照）だとすると，2Mバイト版POLYPHONは65,000円と，ちょっと高めに感じられるかもしれない。ちなみに4Mバイト拡張RAMは59,800円（ツクモ特価）だ。しかし，POLYPHONに載っているRAMはそれだけではない。ボード上のプロセッサ用に2MバイトのRAMが別に載っているのである。ボード用なんてあっても関係ないと思うのはちと早い。POLYPHONにはこれを本体側のRAMディスクとして使えるドライバが付属しているのだ。すると2Mバイトの増設RAM，2MバイトのRAMディスク……，単純にRAMボードとして見ても破格の値段なのがわかるだろう。ついでに，POLYPHONに使われているメモリはSIMMなので，2Mバイト版でも簡単に8Mバイト版に拡張できる。

SIMMというのはMacintoshやIBM PCなどで使用されているメモリモジュールで，秋葉原あたりでは4Mバイトあたり12,000～20,000円で入手できるのだ。



POLYPHON　ネオコンピュータシステム　☎03(5682)7007

で，もちろんPOLYPHONは単なるRAMボードではない。ボード上にはTMP68303が載っている。これはX68000で使用されている68HC000を中心にDMAやシリアル通信コントローラを1チップにまとめた石で，ソフト的には68000CPUとみなしてかまわない。要するにボード上に16MHzの68000が載っているわけだ。

よくあるアクセラレータのように，これを使えば古いマシンでもXVIと同じ速度になる……わけではない。ボード上のCPUは本体のグラフィックやスプライトを直接制御することはできない。その代わり，本体のCPUと並行してプログラムを実行できる。POLYPHONはPCM8などの音楽用のドライバを実行するために設計されたものだが，なにも音楽用のプログラムしか走らないわけではない。サンプルではポリゴンのデモもついている。ちなみにソースプログラムはすべて公開する方針だそうだ。

ボードの位置付けはV70ボードに少し似ているが，こちらは勝手知ったる68000CPUなのだ。

## POLYPHONプログラミング

POLYPHON（以下サブボード）に搭載されているTMP68303はクロック16MHzである。ご存じのとおりXVI以前に発表されてきたX68000のMPU68000はクロック10MHzである。

11月末現在サブボード対応アプリケーションとして，PCM8，RCシステム，MLDミュージックドライバが用意されている。PCM8は本誌92年6月号の付録ディスクに収録されたこともある，ソフトウェアで同時に8声までのPCMを重ねて出力するその筋のツールである。さすがに10MHzのX68000でたくさんのPCMを重ねると処理が重くなることがある。が，サブボードで動かせば，10MHzのマシンでも鼻歌交じりでスキップしているかのような余裕のある処理になる。う～ん素晴らしい。

ただしcommand.x上の実行形式のプログラム（*.x,*.r,*.z）をサブボードに転

送しただけではプログラムの実行はできない。サブボードで動かすには改めてプログラムを作成しなければならない。ちょっと面倒だけどしかたないか。ソフトウェアレベルで68000とTMP68303は互換性があるので，アセンブラを理解しているのなら開発も難しくないだろう。

私としてはMAGICを移植したこともあり，処理速度を稼ぎたい3次元座標から2次元座標への変換処理をサブボードで動かしたらどうだろう……と非常に興味を持った。というわけで今回初めてサブボードで動作するプログラムを書いてみた。読者諸氏のプログラム作成の参考になれば幸いである。

## サブボードプログラミング概略

前述したようにMAGICではワイヤーフレーム表示をするのに，処理時間の大部分を3次元-2次元座標への変換処理とライン表示に取られている。3D-2D変換処理をサブボードで実行させるとして，プログラムの流れは，

1)　メイン側で座標変換に必要なパラメータを取得し，これをサブ側へ転送する
2)　サブ側では送られてきたパラメータをもとに3D-2D変換処理を行う。求めた2D座標をメイン側へ転送する
3)　メイン側でサブ側から送られてきた2D座標を受け取り，画面に描画する

大まかにはこういった流れになる。これ以前にサブボード占有処理を行い，サブボード上に3D-2D変換処理ルーチンを転送しておく必要がある。さらに使用後はサブボードの解放処理がある。

上の処理の例に限らず，メインとサブでMPU間通信が必要になってくる場合がある。これにはFIFOメモリを使う。FIFOとはFirst In First Outの頭文字を取ったもので，最初に送ったデータが受け取り側でも最初に取り出されるバッファ形式のことである。FIFOの大きさは4096バイトある。

メイン側から見たFIFOのI/Oアドレスは$EFF800～$EFF83B(ボードのジャンパ設定でID＝0が指定されている場合。BASEアドレスはID＝0なら$EFF800，ID＝1なら$EFF880)，サブ側から見たFIFOのアドレスは$A00000～$BFFFFFに割り当てられている(ワードアクセス専用)。この範囲ならどのアドレスを使っても動作は同じである。

FIFOはmovemを使って60バイト一括読み込み/書き込みができるのだが，movemは読み込み終了後に1ワードの余計なリードサイクルが入ってしまうので注意が必要である。このサイクルの読み出し元がFIFOになっていると，結果としてFIFOのデータを1ワード失うことになってしまうからだ。したがって読み出し最終アドレスをメイン側なら$EFF83A，サブ側なら$BFFFFEに調整しなくてはいけない。

例：

```
 (メイン側，FIFOから16バイト読み込む)
*  読み込みデータがあることを確認後
   lea.l  $EFF83C-8*2,a0
   movem.w (a0),d0-d7
 (サブ側，FIFOから16バイト読み込む)
*  読み込みデータがあることを確認後
   lea.l  $C00000-8*2,a0
   movem.w (a0),d0-d7
```

転送の処理が，

```
   movem.w (a0)+,……
```

のようになっていないことに注目。FIFOバッファは同じアドレスから読み出すたびに自動的に新しいデータが出てくるのだ。

メイン側とサブ側のプログラムは並列動作が可能だ。サブボードにはサブ側からメイン側へ割り込み要求が1本，メイン側からサブ側への割り込み要求が3本ある。そこで並列動作をさせる場合は割り込みを使って，

1)　メイン側はFIFOを使ってサブ側へデータを渡したあとも処理を続行
2)　サブ側はメイン側から送られてきたパラメータをもとに処理
3)　処理したデータをメイン側へ返す。データをFIFOに書き込んだ直後，メイン側へ割り込み要求を送信する
4)　メイン側の割り込みルーチンでFIFOからデータを取り込む

のようにプログラムしておくと効率よくメイン側のMPUを使える。ただしあまりに頻繁にサブ側からメイン側へ割り込みをかけると，メイン側の処理速度が落ちる。ある程度データをためてから，割り込み要求を送信して一括処理するようにするといいだろう。

サブボードで動作するプログラムを書くならFIFOと割り込み処理は最低押さえておきたい。ほかに大量のデータを扱うならDMAを知らないといけない。

それからこれは重要なことだが，サブ側から本体のメモリ（メインメモリ，グラフィックRAMなど）にはアクセスできない。MAGICなら座標変換処理をサブ側で行い，実際の描画はメイン側で行うようにプログラムしなくてはいけない。

## 実際のプログラミング

実際にサブボード上で動作するプログラムを書いてみたのがリスト1である。説明とあわせて参照してほしい。プログラムの動作は画面にPICなど65536色モードでグラフィックを表示して（表示されている状態でなくてもメモリにあれば大丈夫）実行すると，ある領域を40度回転したものを描画する。サンプルなので実用的には使えないソフトである。ま，今回はサブボードのプログラミング説明ということなのでご勘弁いただきたい。

なお画面回転プログラムは『X68000マシン語グラフィックス編』に掲載されていたBASICプログラムを参考にさせていただ

図1　POLYPHON(68303)のメモリマップ

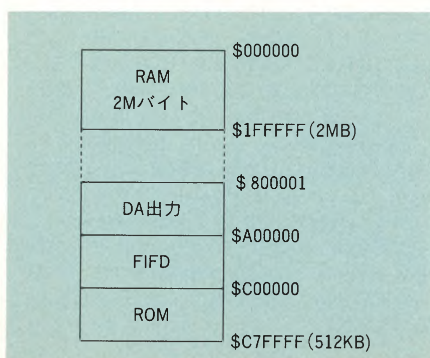
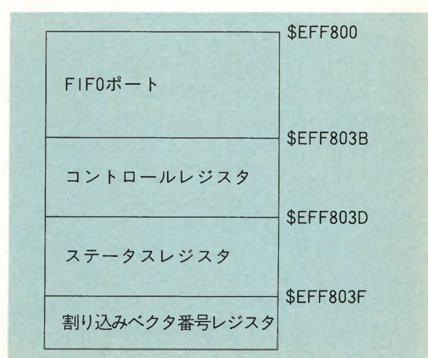



図2　X68000から見たメモリマップ



いた。プログラムはかなり冗長で，高速化してもいない（2倍するのにmulsを使ってたりする）。

プログラムは，メイン側はサブ側へ回転角度の受け渡しと描画，サブ側は回転計算処理を受け持つようにした。全体の流れを大まかに述べると，サブボードの実装検査，占有を行ったあと，メイン側からサブ側へプログラムを転送。メイン側では回転角度をFIFOへ書き込んで，サブ側からの割り込み要求を待つ。サブ側ではこれを受け取り，回転後の座標が回転前にどの座標にあるのか計算して，その座標からG-RAMのアドレスを算出，FIFOに書き込んだのちメイン側へ割り込み要求を送信する。

メイン側の割り込みルーチンはダミー。割り込みがかかるとペンディング状態が0（ない）→1（ある）になるので，メインルーチンでペンディング状態を監視しておいて割り込み要求が発生したか判定する。サブ側ですべての処理が終わったら終了コードをメイン側へ転送する。それを受け取ったメイン側はサブボードの解放処理を行いプログラムを終了する，といった感じである。

もうちょっと細かく見ていこう。最初にサブボードがX68000に実装されているか確認する。リストではサブルーチンsub_board_chkで処理している。実装検査はバスエラーベクタをフックして行う。まず$EFF83Cをワードで読み出せるか試す。サブボードがID0（BASEアドレス$EFF800）にある場合はワードで読み出すことができる。ID1（BASEアドレス$EFF880）に設定している場合はバスエラーが発生してしまうが，あらかじめバスエラー処理ルーチンのエントリをID1の実装チェックルーチンに書き換えてあるので，「バスエラーが発生しました攻撃」はない。先の検査でバスエラーが発生したなら$EFF8BCをワードで読み出してみる。再びバスエラーが発生したらサブボードは実装されていないと判断できる。ワードで読み出すことができた場合は，同じアドレスを今度はバイトで読み出してみる。サブボードが実装されていればバスエラーが発生するはずなので，バスエラーベクタのエントリを適切なアドレスにフックしておくことを忘れないように。

サブボードの実装が確認できたらバスエラーベクタを元に戻して，次にサブボードの占有検査を行う。割り込みベクタアドレス$F8～$FFのエントリアドレスの手前8バイトの内容を調べて，“X68P8S0”，または“X68P8S1”ならサブボードは使用中である。なお末尾の“0”と“1”はサブボードのID番号と同じ数値のはずである。

このチェックを通ったら，次にベクタ番号レジスタ（BASE+$3F）を読んで，$F8以上ならサブボードを使用することができる。$F8未満の場合はリセット状態（BASE+$3Dのbit7が0）なら使用可能。そうでない場合は，メイン側のリセットに対して不揮発なタスクが常駐していると考えられるので使用できない（排除してもメイン側の動作には影響ないが，少なくとも無条件に可としないようにする）。

使用できる場合は，自分用のベクタエントリを$F8～$FFのうちの空きベクタに設定，そのベクタ番号をベクタ番号レジスタに設定して占有処理を終了する。ここで設定するエントリの手前8バイトは“X68KP8Sn”（nはボードID）とし，そのさらに手前の24バイトにタスク名（ASCIIZ形式）を設定する。ベクタエントリの設定は割り込みを使わない場合でも，ダミーのエントリを用意して必ず行う。変更したベクタアドレスと変更前のエントリは，サブボードの解放処理で必要なのでワークエリアに保存しておくこと。

実装確認と占有処理を終えたら，サブボード上のRAMにプログラムを転送する。RAMは$000000～$200000に配置されている。すでに触れたように，サブボードとMPU間の通信はFIFOを使って行う。FIFOを通してサブボード上に転送したデータは，0番地から順番に置かれていく。プログラム転送処理の流れは，

1) FIFOが書き込み可能になるまで待つ
2) FIFOに書き込む（必ずワード単位で）
3) FIFOが読み出し可能になるまで待つ
4) FIFOから読み込む
5) 2)で書き込んだデータと4)で読み込んだデータが同じか検査する
6) 5)の結果が違うなら転送エラー。同じなら1)からプログラムサイズ÷2（ワードだから2で割る）だけ繰り返す。
7) 転送したプログラムのサイズが4096バイトに満たない場合，ダミーデータをFIFOに書き込んで不足分を補う。
8) 4096バイトFIFOに書き込むと，サブボード上に転送したプログラムが自動実行される。

4KバイトFIFOに書き込んだ時点で，サブボード上のプログラムが自動的に実行されるので，4Kバイトを超えるプログラムをサブボード上に転送するのはちょっと面倒である。まずサブボード上で動作するFIFOからデータを読み込むプログラム（いわゆるローダ）を上の手順で転送して実行させる。ローダは自分自身をサブボード上のRAMの最終アドレス近くに転送して実行するようにプログラムしておき，RAMの

## 音楽用途でのPOLYPHON

本文では「音楽用途だけではない」ことの説明のため，音楽用で使う場合の解説がそっくり抜けているので簡単に補足する。

まず，PCM8の場合。

POLYPHONを使う場合，PCM8の処理をボード上のCPUで行い，加工したデータを本体に渡して演奏することができる。といっても，ボード上に特殊なハードがあるわけではなく，68303CPUですべての処理を行う。別名「PCM8ボード」とも呼ばれるゆえんである。

たとえば，PCM8を使っている音楽ドライバがあった場合，普通のPCM8に換えてPOLYPHON用PCM8を組み込めば，まったく問題なく動作する。音楽ドライバはまったく変更する必要はない。ただし，PCM8ドライバを使わずにAD PCMを多声化しているソフトではPOLYPHONの恩恵を受けることはできない。

いくらPCM8とはいえ，AD PCMでは音質や表現力に限界がある。そこで，POLYPHONにはPCMの直接出力端子がついている。POLYPHON上に専用のドライバを転送すればX68000にPCM音源が拡張される。これは8ビットPCMで，もちろん音階つき。PCM8ではできなかった各音のステレオ指定も可能である。これも特殊なハードは使わず68303CPUで処理している。

そして，それらで使用するPCMデータも付属する。Z-MUSICなどでもサンプリングファイルが結構巨大になりがちだが，POLYPHONでは基本ドラムセットをボード上のROMに内蔵している。現在，このROMが未完成のためPCM部分の詳しいレポートは後日改めて行いたい。なお，ROM以外にも高品位PCM音が用意される予定。

続いてMIDIである。これは68303のシリアルコントローラ部を使ったもので，純正のMIDIボードとは互換性がない。しかし，音楽演奏用のドライバが個別対応していけば問題はない。現在，MLD，RCPシステムほかフリーソフトのドライバがすでに対応している。Z-MUSICは機能フィックス中のためまだ対応していない。ゲームなどでは個別のパッチプログラムによってユーザーレベルでPOLYPHONに対応させつつあるようだ。

$000000から4096バイトの範囲をプログラムで使わないようにしておく。そうしておいてサブ側のローダはFIFOからデータが読み込み可能になるのを待つ。

一方メイン側はサブ側で実行するメインプログラムを引き続きFIFOに書き込む。サブ側ではメイン側でFIFOにデータが書き込まれると読み出し可能になるから，ローダのほうでFIFOから読み込んだデータを$000000から順番に転送するようにしておく。これをプログラムサイズ÷2だけ繰り返せばいいので，メイン側でメインプログラムの転送に先立って転送するサイズをFIFOに書き込んでおき，サブ側に通知するようにすればいいだろう。

リストではラベルa_start～a_endがサブボード上に転送される。サブボード上に転送するプログラムは，プログラムの先頭に初期SP，初期PC，68000のベクタエントリ（62種類），サブボードの内蔵デバイスのベクタエントリ（34種類）を定義しなくてはいけない。特に重要なのは，

$000000.L　実行時のSP初期値
$000004.L　実行アドレス

だ。

内蔵デバイスのベクタアドレスの内容については，よくわからないので付属のPCM8SB.HAS（江藤啓氏作）のソースリストから抜粋させていただいた。

ラベルa_execからはFIFOで使うDMAC，インタラプトコントローラの初期化を行っている。DMACについてはネオコンピュータシステムの方から資料をいただいたのだが，その他の内蔵デバイスについては資料がなく，インタラプトコントローラの設定もPCM8SB.HASのものをそのまま真似て設定している（動いているのでこれでいいんだろう）。

ラベルs_mainからがサブ側のメインルーチンである。最初にFIFOに読み込み可能なデータがくるのを待つ。これはメイン側の処理にある「回転角度をFIFOに書き込む」動作を待つ処理である。FIFOの読み込みが可能となったら，読み出したデータを回転角度とみなして座標計算を行う。座標を求めたらそこからG-RAM上のアドレスを求め，FIFOに書き込む。その直後にFIFOにデータを書き込んだことをメイン側へ知らせるために，割り込み要求を送信する。すべての処理が終わったら終了コードとして－1を送信するようにした。

一方，メイン側では回転角度をFIFOに書き込んだあとはペンディング状態を監視しておき，（BASE＋$3D bit6）が1から0になったらFIFOに書き込みがあったと判断する。ペンディング状態をクリアして，INT0の割り込みをマスクする（マスク中にかかった割り込みはペンディングされる）。FIFOから取り出したG-RAMのアドレスに対して読み込み，書き込みを行ったあとにマスク状態を解除する。マスク解除と同時にペンディング中の割り込みは受け付けられる。エンドコードである－1をFIFOから取り出したらサブボードの解放処理に移る。ラベルsub_board_freeの処理だ。割り込みを禁止して使ったベクタのエントリを使用前の状態に戻す。ベクタ番号レジスタ(BASE＋$3F)に$FFを設定する。

## 最後に

とりあえず作ってみたというのが正直な感想だ。さんざん説明しておいて，いまさらこういうのもなんだが，もしかしたらプログラムの構造にどこかまずいところがあるかもしれない。特に割り込み関係は私自身よくわからない。

まあ動いているんだから大丈夫なんだろう。原稿を書くにあたってわからないことだらけだったのだが，NCS-NET上で江藤啓氏にわからない点を質問し，氏から迅速かつ丁寧な回答をいただいた。なお掲載したリストのうちPOLYPHON.MACは氏が書かれたものだ。氏にはこの場を借りて感謝の意を表したい。

そういえばポリゴン版MAGICがあったんだよな～。

**リスト1　POLYPHON.MAC**

```
.nlist
************************************************************
*
* Project POLYPHOON :-)
* 68303 sub-processor board
*
* メイン/サブMPU用定数
*
************************************************************
* v1.02   92/10/26        written by: H.Etoh
************************************************************

************************************************************
*
* メイン側
*
************************************************************

mP8B0              set $EFF800     * ボード0
mP8B1              set $EFF880     * ボード1
mFIFO              set $EFF800     * FIFO先頭
mFIFOEND           set $EFF83C     * FIFO最終(movem対策)
mCTRL              set $EFF83D     * コントロール
mVECR              set $EFF83F     * ベクタ

mB0                set $00
mB1                set $80
mBF                set $00
mBE                set $3C
mBC                set $3D
mBV                set $3F

mrP8B0             set $00
mrP8B1             set $80
mrFIFO             set $00
mrFIFOEND          set $3C
mrCTRL             set $3D
mrVECR             set $3F

*** ステータスビット番号（全ビット負論理）

mFRE               set 0           * FIFO 読み出し側エンプティ
mFRH               set 1           * FIFO 読み出し側ハーフ
mFRF               set 2           * FIFO 読み出し側フル
mFWE               set 3           * FIFO 書き込み側エンプティ
mFWH               set 4           * FIFO 書き込み側ハーフ
mFWF               set 5           * FIFO 書き込み側フル
mIPE               set 6           * 割込ペンディング
mRST               set 7           * リセット

************************************************************
*
* サブ側
*
************************************************************

*
* 外部デバイスマップ
*

sRAMTOP            set $000000     * RAM先頭
sRAMEND            set $200000     * RAM最終+1
sVOICE             set $800000     * PCM出力
sVOICE_L           set $800001     *    +1 : Lch
sVOICE_R           set $800003     *    +3 : Rch
sFIFO              set $A00000     * FIFO先頭
sFIFOEND           set $C00000     * FIFO最終+1 (movem対策)
sROMTOP            set $C00000     * ROM先頭
sROMEND            set $C80000     * ROM最終+1

*
* 内蔵デバイスマップ
*
* 各レジスタは、絶対ショート形式でアクセス出来ます
* (HAS の -a オプションが便利)
* シンボル名は東芝純正表記の頭に's'を付加で統一
*

sBASE              set $FFFFFC00   * 内蔵デバイスベース

```

```
*** アドレスデコーダ

sAMAR0          set $FFFFFC00   * B: DRAM 先頭アドレス
sAMR0           set $FFFFFC01   * B: DRAM アドレスマスク
sARCR0          set $FFFFFC03   * B: DRAM 使用許可 / DTACK方式 / WAIT数
sAMAR1          set $FFFFFC04   * B: (CS1 先頭アドレス)
sAMR1           set $FFFFFC05   * B: (CS1 アドレスマスク)
sARCR1          set $FFFFFC07   * B: (CS1 使用許可 / DTACK方式 / WAIT数)
sARCR2          set $FFFFFC09   * B: IACK DTACK方式 / ウエイト
sATOR           set $FFFFFC0B   * B: タイムアウト時間
sARELR          set $FFFFFC0C   * W: 内蔵デバイスアドレス

*** インタラプトコントローラ

sICR0           set $FFFFFC81   * B: INT0 割込方式 / レベル
sICR1           set $FFFFFC83   * B: INT1 割込方式 / レベル
sICR2           set $FFFFFC85   * B: INT2 割込方式 / レベル
sICR3           set $FFFFFC87   * B: ｼﾘｱﾙ0 レベル
sICR4           set $FFFFFC89   * B: ｼﾘｱﾙ1 レベル
sICR5           set $FFFFFC8B   * B: DMAC0 レベル
sICR6           set $FFFFFC8C   * W: (DMAC2/1 レベル)
sICR8           set $FFFFFC8F   * B: ﾀｲﾏ1 レベル
sICR9           set $FFFFFC91   * B: ﾀｲﾏ2 レベル
sICR10          set $FFFFFC92   * W: ﾀｲﾏ4/3 レベル
sIMR            set $FFFFFC94   * W: 割込マスク
sIPR            set $FFFFFC96   * W: 割込ペンディング
sIISR           set $FFFFFC98   * W: 割込サービス中
sIVNR           set $FFFFFC9B   * W: 割込ベクタ番号

*** シリアルコントローラ

sSMR0           set $FFFFFD81   * B: ｼﾘｱﾙ0 モード
sSCMR0          set $FFFFFD83   * B: ｼﾘｱﾙ0 コマンド
sSBRR0          set $FFFFFD85   * B: ｼﾘｱﾙ0 ボーレート
sSSR0           set $FFFFFD87   * B: ｼﾘｱﾙ0 ステータス
sSDR0           set $FFFFFD89   * B: ｼﾘｱﾙ0 データ
sSPR            set $FFFFFD8D   * B: ｼﾘｱﾙ0/1 プリスケーラ
sSCR            set $FFFFFD8F   * B: ｼﾘｱﾙ0/1 コントロール
sSMR1           set $FFFFFD91   * B: ｼﾘｱﾙ1 モード
sSCMR1          set $FFFFFD93   * B: ｼﾘｱﾙ1 コマンド
sSBRR1          set $FFFFFD95   * B: ｼﾘｱﾙ1 ボーレート
sSSR1           set $FFFFFD97   * B: ｼﾘｱﾙ1 ステータス
sSDR1           set $FFFFFD99   * B: ｼﾘｱﾙ1 データ

*** パラレルコントローラ ( + ステッピングモータ )

sSMMDR          set $FFFFFF31   * B: ﾎﾟｰﾄ0/1 機能選択
sP0R            set $FFFFFF33   * B: ﾎﾟｰﾄ0 データ
sP1R            set $FFFFFF35   * B: ﾎﾟｰﾄ1 データ
sP01CR          set $FFFFFF37   * B: ﾎﾟｰﾄ0/1 入出力方向
sSMCR           set $FFFFFF39   * B: (ｽﾃｯﾋﾟﾝｸﾞﾓｰﾀ 回転方向)
sP2R            set $FFFFFDB1   * B: ﾎﾟｰﾄ2 データ
sP2CR           set $FFFFFDB3   * B: ﾎﾟｰﾄ2 入出力方向

*** 16ビットタイマ

sTCR0           set $FFFFFE00   * W: (ｳｫｯﾁﾄﾞｯｸﾞ ﾀｲﾏ コントロール)
sTMCR01         set $FFFFFE04   * W: (ｳｫｯﾁﾄﾞｯｸﾞ ﾀｲﾏ 最大カウント)
sTCTR0          set $FFFFFE0C   * W: (ｳｫｯﾁﾄﾞｯｸﾞ ﾀｲﾏ カウンタ)
sTCR1           set $FFFFFE20   * W: ﾀｲﾏ1 コントロール
sTMCR11         set $FFFFFE24   * W: ﾀｲﾏ1 最大カウント1
sTMCR12         set $FFFFFE28   * W: ﾀｲﾏ1 最大カウント2
sTCTR1          set $FFFFFE2C   * W: ﾀｲﾏ1 カウント・レジスタ
sTCR2           set $FFFFFE40   * W: ﾀｲﾏ2 コントロール
sTMCR21         set $FFFFFE44   * W: ﾀｲﾏ2 最大カウント1
sTMCR22         set $FFFFFE48   * W: ﾀｲﾏ2 最大カウント2
sTCTR2          set $FFFFFE4C   * W: ﾀｲﾏ2 カウント・レジスタ

*** 8ビットタイマ

sTIMR30         set $FFFFFE61   * B: ﾀｲﾏ3 タイマ・レジスタ(sub)
sTIMR31         set $FFFFFE63   * B: ﾀｲﾏ3 タイマ・レジスタ(main)
sTIMR40         set $FFFFFE65   * B: ﾀｲﾏ4 タイマ・レジスタ(sub)
sTIMR41         set $FFFFFE67   * B: ﾀｲﾏ4 タイマ・レジスタ(main)
sTCCR           set $FFFFFE69   * B: ﾀｲﾏ3/4 クロック
sTFFCR          set $FFFFFE6B   * B: ﾀｲﾏ3/4 出力コントロール
sTMDR           set $FFFFFE6D   * B: ﾀｲﾏ3/4 モード
sTRCR           set $FFFFFE6F   * B: ﾀｲﾏ3/4 動作コントロール
sTIRCR          set $FFFFFE71   * B: ﾀｲﾏ3/4 割込コントロール

*** DMAコントローラ

sCHCR0          set $FFFFFE80   * W: DMAC0 コントロール
sDTCR0          set $FFFFFE82   * W: DMAC0 転送カウント
sSMAR           set $FFFFFE84   * L: DMAC0 ソースアドレス
sDMAR           set $FFFFFE88   * L: DMAC0 デスティネーションアドレス
sOPCR           set $FFFFFE8D   * B: DMAC0/1/2 動作コントロール
sCHSR           set $FFFFFE8E   * W: DMAC0/1/2 ステータス
sCHCR1          set $FFFFFE90   * W: (DMAC1 コントロール)
sDTCR1          set $FFFFFE92   * W: (DMAC1 転送カウント)
sMAR1           set $FFFFFE94   * L: (DMAC1 メモリアドレス)
sCHCR2          set $FFFFFEA0   * W: (DMAC2 コントロール)
sDTCR2          set $FFFFFEA2   * W: (DMAC2 転送カウント)
sMAR2           set $FFFFFEA4   * L: (DMAC2 メモリアドレス)

*** DRAMコントローラ

sRCR            set $FFFFFF01   * B: リフレッシュ・コントロール
sMCR            set $FFFFFF03   * B: メモリ・コントロール

*** P0/P1 ビット番号

sIRQ            set 0           * P0: 割込送信
sFRE            set 1           * P0: FIFO 読み出し側エンプティ
sFRH            set 2           * P0: FIFO 読み出し側ハーフ
sFRF            set 3           * P0: FIFO 読み出し側フル
sFWE            set 1           * P1: FIFO 書き込み側エンプティ
sFWH            set 2           * P1: FIFO 書き込み側ハーフ
sFWF            set 3           * P1: FIFO 書き込み側フル

*************************************************************

.list
```

## リスト2

```

  .include      doscall.mac
  .include      iocscall.mac
  .include      polyphon.mac

rx:             equ     3
ry:             equ     2

  .text
  .even

start:
        clr.l   -(sp)
        DOS     _SUPER                  * スーパーバイザモードにする
        move.l  d0,(sp)                 * SSP保存

        move.w  #$10c,d1                * 512*512 65536色
        IOCS    _CRTMOD                 * 画面の初期化をしない
w_a:    btst.b  #4,$e88001
        bne     w_a
        move.b  #$2f,$e82601            * グラフィック画面表示

        move.w  #2,-(sp)
        move.w  #10,-(sp)
        DOS     _CONCTRL                * 画面全体をクリア
        addq.l  #4,sp

        lea.l   irq,a0                  * 割り込みアドレス
        bsr     sub_board_chk           * サブボードがあるか調べる
        tst.l   d0
        bne     error
                                        * d0=0ならサブボードは実装されている
skip:
        move.l  a0,use_vect_adrs        * 直前のベクタアドレスを保存
        move.l  a1,old_vect_entry       * 直前のベクタエントリを保存

        bsr     sub_load                * サブボードにロード

        lea.l   mFIFOEND-8,a0           * movem対策
w_b:    btst.b  #5,mrCTRL(a5)
        beq     w_b
        move.w  #80,mFIFO               * FIFOに回転角度を書き込む

wait_sub_work:
        btst.b  #mIPE,mrCTRL(a5)
        bne     wait_sub_work           * 割込ペンディング中
        move.b  mrVECR(a5),mrVECR(a5)
        move.b  #$81,mrCTRL(a5)         * サブ側に割り込み要求送信
        movem.w (a0),d4-d7
        cmpi.w  #-1,d4
        beq     sub_board_free
        swap    d4
        move.w  d5,d4
        swap    d6
        move.w  d7,d6
        movea.l d4,a2
        movea.l d6,a3
        move.b  #$80,mrCTRL(a5)
        move.w  (a2),(a3)
        bra     wait_sub_work

*********************************************************
* 終了処理
*********************************************************
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++
sub_board_free:
        move.w  sr,-(sp)
        ori.w   #$700,sr
        move.l  use_vect_adrs,a0
        cmpi.l  #irq,(a0)
        bne     free_err
        move.l  old_vect_entry,(a0)
        moveq.l #$80,d0
        and.b   d0,mrCTRL(a5)
        move.b  #$ff,mrVECR(a5)
        move.w  (sp)+,sr
end:
        DOS     _SUPER                  * ユーザーモード復帰
        addq.l  #4,sp
        DOS     _EXIT

*********************************************************
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++
error:
        bpl     busy
        lea.l   err_mes,a1              * サブボードは実装されていない
        bra     abort
busy:
        lea.l   busy_mes,a1             * サブボードは既に使用されている
        bra     abort
free_err:
        lea.l   free_mes,a1             * サブボードの解放処理に失敗した
        bra     abort
trans_error:
        lea.l   trans_mes,a1
abort:
```



▶ 僕はアヴァンポップなX68000用ゲームを作りたいと思っているので，12月号の知能機械概論を興味深く読みました。ちなみに「ねこじるうどん」は「ねこぢるうどん」といい，漫画雑誌「ガロ」に現在も連載されている人気漫画です。　野口　友則(20)茨城県

```
            pea.l   (a1)
            DOS     _PRINT
            addq.l  #4,sp
            bra     end
************************************************************
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
sub_load:
            move.b  #0,mrCTRL(a5)               * サブボードリセット
            moveq.l #$3f,d0
sb_init_wait:
            dbra    d0,sb_init_wait             * ディレイ
sb_init_loop:
            move.b  #$80,mrCTRL(a5)             * サブボードリセット解除
            moveq.l #$3f,d0
sb_init_wait2:
            dbra    d0,sb_init_wait2            * ディレイ
            tst.b   mrCTRL(a5)
            bpl     sb_init_loop
check_irq:
            btst.b  #6,mrCTRL(a5)
            bne     check_irq
            move.b  mrVECR(a5),mrVECR(a5)
*
* プログラムロード
*
            lea     a_start(pc),a0              * サブプログラム格納先頭アドレス
            move.w  #((a_end-a_start)/2)-1,d0          * サブプログラム長 -1
wait_write:
            btst.b  #5,mrCTRL(a5)
            beq     wait_write                  * 書き込みFIFOが4096バイト埋まっている
            move.w  (a0)+,d1
            move.w  d1,(a5)                     * 1ワードFIFOに書き込む
verify:
            btst.b  #0,mrCTRL(a5)
            beq     verify                      * 読み出しFIFOが空だ
            cmp.w   (a5),d1
            dbne    d0,wait_write
            bne     trans_error                 * 書き込みに失敗した
            moveq.l #-1,d1                      * 4096バイトに足りない部分を$FFFFで埋める
            move.w  #4095-((a_end-a_start)/2),d0
wait_write2:
            btst.b  #5,mrCTRL(a5)
            beq     wait_write2
            move.w  d1,(a5)
verify2:
            btst.b  #0,mrCTRL(a5)
            beq     verify2
            cmp.w   (a5),d1
            dbne    d0,wait_write2
            bne     trans_error                 * 書き込みに失敗した
w_c:        btst.b  #6,mrCTRL(a5)               * check irq
            bne     w_c
            move.b  mrVECR(a5),mrVECR(a5)       * ペンディングクリア
            nop
            move.b  #$81,mrCTRL(a5)             * サブ側に割り込みを要求
            nop
            nop
            nop
            nop
            nop
            move.b  #$80,mrCTRL(a5)
            rts
************************************************************
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
*           '...|...|...|...|...|...  |
            .dc.b   'POLYPHON SAMPLE PROGRAM',0
            .dc.b   'X68KP8S0'
irq:
  rte
************************************************************
* in a0.l           割り込みエントリアドレス
*
* out d0.l = 0      サブボードは使える
*          = -1     サブボードは挿入されていない
*          = 1      サブボードは使われている
* a0.l              使用するベクタアドレス
* a1.l              直前のベクタエントリ
* a5.l              サブボードがマッピングされている先頭アドレス
*----------------------------------------------------------
* サブボードが実装されているか調べる
*----------------------------------------------------------
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
sub_board_chk:
            move.w  sr,-(sp)        * SRを退避
            ori.w   #$700,sr        * 割り込み禁止
            lea.l   mP8B0,a5        * $eff800にマッピングされているか?
            move.l  sp,a1           * SPを退避
            move.l  8.w,a2          * バスエラー処理ルーチンを退避
            move.l  #sub_chk1,8.w   * バスエラー処理ルーチンを変更
            tst.w   mrCTRL-1(a5)    * 読み込めるか?
            nop
            move.l  #'P8S0',d1
            bra     sub_chk_ok      * サブボードは$eff800~$eff83f
sub_chk1:
  lea.l     $80(a5),a5              * $eff880にマッピングされているか?
            move.l  #sub_err,8.w    * バスエラー処理ルーチンを変更
            tst.w   mrCTRL-1(a5)    * 読み込めるか?
            nop
            move.l  #'P8S1',d1
            bra     sub_chk_ok      * サブボードは$eff880~$eff8bf
sub_chk2:
            move.l  #sub_err,8.w    * バスエラー処理ルーチンを変更
            tst.b   mrCTRL-1(a5)    * サブボードがあればバスエラーになる
sub_chk_ok:
            move.l  a1,sp           * SP復帰
            move.l  a2,8.w          * バスエラー処理ルーチン復帰
*----------------------------------------------------------
* サブボードが使えるか調べる
*----------------------------------------------------------
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* d4.l = work register
            lea.l   $ff*4+4,a1        * 割り込みベクタ$ff+4のアドレス
                                      * (後でプリデクリメントするから)
            moveq.l #0,d2             * 未使用のベクタ番号を格納するためのワーク
            moveq.l #$ff,d3           * 未使用のベクタ番号を求めるためのワーク
            moveq.l #8-1,d0
busy_check_loop:
            move.l  -(a1),a2
            cmpi.l  #'X68K',-8(a2)  * エントリアドレスの手前8バイトから
            bne     busy_check_next * 'X68KP8Sn'(nはボードID)
            cmp.l   -4(a2),d1         * があればサブボードは使用中
            beq     sub_board_busy
busy_check_next:
            tst.b   d2
            bne     busy_check_skip * 既に未使用のベクタ番号を確保した
            move.l  a2,d4
            cmpi.l  #$1000000,d4      * 68000では有効ビット幅が24ビットなので
            bcs     busy_check_skip * ベクタエントリが24ビット以下なら
                                      * そのベクタは使用されていると考える
            movea.l a1,a3             * 未使用のベクタエントリ
            move.b  d3,d2             * 未使用のベクタ番号
busy_check_skip:
            addq.b  #1,d3
            dbra    d0,busy_check_loop
            tst.b   d2
            beq     sub_board_busy  * 未使用のベクタが見つからなかった
            cmpi.b  #$f8,mrVECR(a5)
            bcc     sub_board_ok      * 割り込みベクタ番号が$f8以上なら使える
            btst.b  #7,mrCTRL(a5)
            bne     sub_board_busy
sub_board_ok:
            move.b  d2,mrVECR(a5)
            movea.l (a3),a1           * 直前のベクタエントリ
            move.l  a0,(a3)
            move.l  d1,-(a0)          * ID(P8Sn)を書き換える
            move.l  #'X68K',-(a0)
            movea.l a3,a0             * 使用するベクタアドレス
            move.w  (sp)+,sr          * SR復帰
            moveq.l #0,d0             * サブボードは使える
            rts
sub_err:
            move.l  a1,sp             * SP復帰
            move.l  a2,8.w            * バスエラー処理ルーチン復帰
            move.w  (sp)+,sr          * SR復帰
            moveq.l #-1,d0            * サブボードは挿入されていない
            rts
sub_board_busy:
            move.w  (sp)+,sr          * SR復帰
            moveq.l #1,d0             * サブボードは使われている
            rts
************************************************************
* サブボード側のプログラム
************************************************************
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
a_stack     equ     a_end+$8000       * スタックアドレス
a_start:
            .dc.l   a_stack                     * initial stack
            .dc.l   a_exec-a_start              * reset
            .dc.l   -1,-1                       * w-fault for bus/adr err
            .dcb.l  60,a_rte-a_start            * no operation for others
************************************************************
* INTERNAL DEVICE VECTORS
*
            .dc.l   a_ext_0-a_start             * External 0
            .dc.l   a_ext_1-a_start             * External 1
            .dc.l   a_ext_2-a_start             * External 2
            .dc.l   a_rte-a_start               * x
            .dc.l   a_timer_1-a_start           * Timer 1
            .dc.l   a_timer_2-a_start           * Timer 2
            .dc.l   a_timer_3-a_start           * Timer 3
            .dc.l   a_timer_4-a_start           * Timer 4
            .dc.l   a_tx_irq_0-a_start          * Serial 0 Tx buff empty
            .dc.l   a_rx_irq-a_start            * Serial 0 Rx buff full
            .dc.l   a_rx_err-a_start            * Serial 0 Rx err
            .dc.l   a_rte-a_start               * x ( Serial 0 Special case )
            .dc.l   a_tx_irq_1-a_start          * Serial 1 Tx buff empty
            .dc.l   a_rte-a_start               * x ( Serial 1 Rx buff full )
            .dc.l   a_rte-a_start               * x ( Serial 1 Rx err )
            .dc.l   a_rte-a_start               * x ( Serial 1 Special case )
            .dc.l   a_dma_err-a_start           * DMAC 0 Err
            .dc.l   a_dma_irq-a_start           * DMAC 0 Service end
            .dc.l   a_rte-a_start               * x ( DMAC 1 Err )
            .dc.l   a_rte-a_start               * x ( DMAC 1 Service end )
            .dc.l   a_rte-a_start               * x ( DMAC 2 Err )
            .dc.l   a_rte-a_start               * x ( DMAC 2 Service end )
            .dc.l   a_rte-a_start               * x
            .dc.l   a_rte-a_start               * x
            .dc.l   a_rte-a_start               * x
            .dc.l   a_rte-a_start               * x
            .dc.l   a_rte-a_start               * x
            .dc.l   a_rte-a_start               * x
            .dc.l   a_rte-a_start               * x
            .dc.l   a_rte-a_start               * x
            .dc.l   a_rte-a_start               * x
            .dc.l   a_rte-a_start               * x ( DEFAULT )
************************************************************
* DUMMY INTERRUPT
*
```

▶ なんと！　１月号の予告に「THUNDER WORD」とあるではないですか。これは，1990年９月号（こんなに古かったのか）で姿を消した，祝一平氏による「雷語１号」では？　もしそうならどんなワープロだろう。期待がふくらんでしまう，わくわく。

前田　光章(20)静岡県

```
a_ext_0:
a_ext_1:
a_ext_2:
a_timer_1:
a_timer_2:
a_timer_3:
a_timer_4:
a_tx_irq_0:
a_tx_irq_1:
a_rx_irq:
a_rx_err:
a_dma_irq:
a_dma_err:
a_rt
        rte

a_exec:
*
* DEVICE INIT
*
        move.w  #$2700,sr
        lea.l   sFIFO,a5
        move.w  #0,sDTCR1.w             * DMAC1 転送カウント
        move.w  #%0111_11_1110,sCHCR1.w * DMAC1 コントロール 受信
        move.w  #0,sDTCR2.w             * DMAC2 転送カウント
        move.w  #%0111_11_1010,sCHCR2.w * DMAC2 コントロール 送信
        move.b  #$31,sICR0.w    * int0 rising edge / level 1
        move.w  #$fffe,sIPR.w   * int0 pending clear
        move.w  #$2000,sr
*
* MAIN
*
        bset.b  #sIRQ,sP0R.w    * 割り込み要求
        bclr.b  #sIRQ,sP0R.w

w_d:    btst.b  #sFRE,sP0R.w
        beq     w_d             * FIFOにデータが書き込まれるのを待つ
        move.w  (a5),d7         * 回転角度

        lea.l   data,a4
        lea.l   $c00000,a0
s_main:
        move.w  d7,d1
        bsr     coshl
        move.w  d1,-(sp)
        move.w  d7,d1
        bsr     sinhl
        move.w  (sp)+,d2        * d2 = cos
        move.w  d1,d3           * d3 = sin

        move.l  #224-1,d4       * dy
s_l1:   move.l  #192-1,d5       * dx
s_l2:   move.w  d2,d7
        move.w  d3,d6
        move.w  d5,d0
        subi.w  #64,d0
        muls    d0,d7
        asl.l   #2,d7
        swap    d7
        muls    #rx,d7
        divs    #ry,d7
        move.w  d4,d0
        subi.w  #64,d0
        muls    d0,d6
        asl.l   #2,d6
        swap    d6
        sub.l   d6,d7
        muls    #ry,d7
        divs    #rx,d7
        addi.w  #128,d7
        move.w  d7,a1                   * x0

        move.w  d2,d7
        move.w  d3,d6
        move.w  d5,d0
        subi.w  #64,d0
        muls    d0,d6
        asl.l   #2,d6
        swap    d6
        muls    #rx,d6
        divs    #ry,d6
        move.w  d4,d0
        subi.w  #64,d0
        muls    d0,d7
        asl.l   #2,d7
        swap    d7
        add.l   d6,d7
        addi.w  #128,d7                 * y0

        move.w  a1,d6
        moveq.l #10,d0
        lsl.l   d0,d7
        add.w   d6,d6
        add.w   d6,d7
        add.l   a0,d7
        move.l  d7,(a4)+
        move.l  #288-1,d6
        move.l  #128-1,d7
        add.w   d5,d6
        add.w   d4,d7
        lsl.l   d0,d7
        add.w   d6,d6
        add.w   d6,d7
        add.l   a0,d7
        move.l  d7,(a4)+

w_e:    btst.b  #3,sP1R.w
        beq     w_e             * FIFOに書き込み可能になるまで待つ
        lea.l   data,a4
@@:
        move.w  (a4)+,(a5)      * FIFOに書き込む
        move.w  (a4)+,(a5)      *       .
        move.w  (a4)+,(a5)      *       .
        move.w  (a4)+,(a5)      *       .
        bset.b  #sIRQ,sP0R.w    * メイン側へ割り込み要求送信
        bclr.b  #sIRQ,sP0R.w
        lea.l   data,a4
s_skip:
        dbra    d5,s_l2
        dbra    d4,s_l1

w_f:    btst.b  #sFWH,sP1R.w
        beq     w_f             * FIFOが書き込み可能になるまで待つ
        move.w  #-1,(a5)        * 終了コードを送る
        move.w  #-1,(a5)
        move.w  #-1,(a5)
        move.w  #-1,(a5)
w_g:    bset.b  #sIRQ,sP0R.w    * メイン側へ割り込み要求送信
        bclr.b  #sIRQ,sP0R.w
        bra     w_g

*********************************************************
* coshl
* in    d1.w = 角度
* out   d1.w = cos(d1.w)
*********************************************************
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++
coshl:
        addi.w  #90,d1
*********************************************************
* sinhl
* in    d1.w = 角度
* out   d1.w = sin(d1.w)
*********************************************************
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++
sinhl:
        subi.w  #360,d1
        bpl     sinhl
sinhl2:
        addi.w  #360,d1
        bmi     sinhl2

        move.w  #180,d2
        sub.w   d2,d1
        bpl     sinhl3
        add.w   d2,d1
sinhl3:
        scs     d0
        cmpi.w  #90,d1
        bmi     sinhl4
        sub.w   d1,d2
        move.w  d2,d1
sinhl4:
        add.w   d1,d1
        move.w  sin_table(pc,d1.w),d1
        tst.b   d0
        bne     sinhl5
        neg.w   d1
sinhl5:
        rts

sin_table:
        dc.w    $0000,$011d,$023b,$0359
        dc.w    $0476,$0593,$06b0,$07cc
        dc.w    $08e8,$0a03,$0b1d,$0c36
        dc.w    $0d4e,$0e65,$0f7b,$1090
        dc.w    $11a4,$12b6,$13c6,$14d6

        dc.w    $15e3,$16ef,$17f9,$1901
        dc.w    $1a07,$1b0c,$1c0e,$1d0e
        dc.w    $1e0b,$1f07,$2000,$20f6
        dc.w    $21ea,$22db,$23c9,$24b5
        dc.w    $259e,$2684,$2766,$2846

        dc.w    $2923,$29fc,$2ad3,$2ba5
        dc.w    $2c75,$2d41,$2e09,$2ece
        dc.w    $2f8f,$304d,$3106,$31bc
        dc.w    $326e,$331c,$33c6,$346c
        dc.w    $350e,$35ac,$3646,$36db

        dc.w    $376c,$37f9,$3882,$3906
        dc.w    $3985,$3a00,$3a77,$3ae9
        dc.w    $3b56,$3bbf,$3c23,$3c83
        dc.w    $3cde,$3d34,$3d85,$3dd1
        dc.w    $3e19,$3e5c,$3e99,$3ed2

        dc.w    $3f07,$3f36,$3f60,$3f85
        dc.w    $3fa6,$3fc1,$3fd8,$3fe9
        dc.w    $3ff6,$3ffd,$4000
data:
        ds.l    2

a_end:

        .data

use_vect_adrs:
        ds.l    1
old_vect_entry:
        ds.l    1
end_flg:
        dc.w    0
err_mes:
        dc.b    'POLYPHONは拡張スロットに挿入されていないようです',13,10,0
busy_mes:
        dc.b    'POLYPHONは既に使われています',13,10,0
free_mes:
        dc.b    'ベクタが書き換えられています。'
        dc.b    'サブボードの解放処理ができませんでした',13,10,0
trans_mes:
        dc.b    'サブボードへのプログラム転送に失敗しました',13,10,0

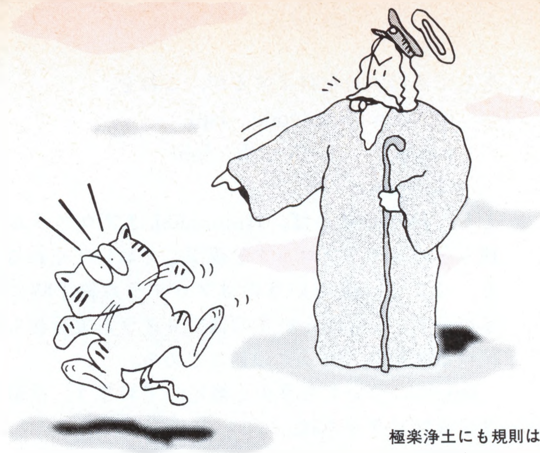
        .end
```

吾輩はX68000である

［第19回］

# 極楽た～ぼマウスII

Izumi Daisuke
泉　大介



極楽浄土にも規則はある
きっちり守って
楽しく使おう

極楽た～ぼマウスの使い心地はいかがだろうか。前回は，た～ぼマウスのプログラムがメモリのどこからどこまでを使っているのかをHuman68kに知らせておかなければ，別のプログラムによって壊されてしまう可能性がある。というところで誌面が尽きてしまった。今回は，ここから話を続けることにしよう。

## ◆Human68kのメモリ管理

OSの大切な役割のひとつとして，コンピュータの資源を管理することがある。とりわけ複数のプログラムが同時に実行されるマルチタスクOSでは，2つ以上のプログラムが同時に同じファイルにデータを書きにいかないか，メモリの同じ場所を同時にワークに使ったりしないかなどということを，常に見張っている必要がある。この管理を楽にするために，これらのOSではユーザーの作成したプログラムは「○○バイトのメモリがほしい」とOSに要求してワークエリアを確保する，という方法がとられている。

プログラムを実行するときも同様で，シェルがそのプログラムのために「○○バイトのメモリがほしい」とOSに要求し，割り当てられたメモリにプログラムを読み込んで実行してくれるようになっている。つまりユーザーの作成したプログラムは，もはやコンピュータを自由自在に扱える唯一無二の絶対者ではなく，OSというお釈迦さまの掌の上で遊んでいる孫悟空に過ぎない存在になってしまっているわけだ。割り当てられたメモリを超えてデータを書き込もうとすると，たちまち緊箍児(きんこじ)は頭を締めつけ，仏様の怒りの霆(いかずち)が打ち降ろされることになる。

吾輩のOSたるHuman68kの場合は，シングルタスクのOSということもあって制約はこれほど厳しくはない。しかし，チャイルドプロセスによって同時に2つ以上のプログラムがメモリ上に存在しうる以上，ある程度のメモリ管理能力はHuman68kにも必要だ。既存のプログラムに重なるアドレスに，チャイルドプロセスで起動したプログラムが読み込まれるようなことがあっては，諸兄も安心してプログラムを実行することはできまい。

そのメモリ管理方法だが，Human68kはメモリを図1-1のような形式でプログラムに割り当てるようになっている。頭にはメモリ管理テーブルと呼ばれるロングワード4つ分のデータ，続いてプロセス管理テーブルと呼ばれる240バイトのデータがあり，そのあとに諸兄のお作りになったプログラムなどが収められる領域が続く。今回はプロセス管理テーブルの内容は必要ないので，図1-1ではメモリ管理テーブルだけをクローズアップしてある。メモリ管理テーブルの先頭と最後のデータがメモリのリンケージを作り出し，こうしてHuman68kは図1-2のようにブロックのつながりとしてメモリを管理しているのである。

諸兄がプログラムを実行なさるときには，この形式に従って空きメモリからメモリブロックが切り出されて，そのプログラムに割り当てられる。プログラムの実行終了と同時にこのメモリブロックは再び解放され，空きメモリに戻されるのである。デバッガのA，ANコマンドでメモリの適当な場所に作成したプログラムは，いわばこの空きメモリの中に勝手に作ったプログラムであり，Human68kはその存在をまったく関知していない。つまり，デバッガを終了してほかのプログラムを起動するときに，徴収される可能性のあるメモリなのである。そんなところへ割り込み処理プログラムを配置するのは自殺行為もいいところ。割り込みがかかると同時にほかのプログラムの内部へ突入し，あらぬ動作を始めることになってしまう。

た～ぼマウスのプログラムで使用しているメモリが，ほかのプログラムによって壊されないようにするためには，Human68kのこのメモリ管理方式に従った方法でた～ぼマウスのためにメモリを確保し，さらに，プログラムの実行が終了してもそのメモリブロックが解放されないようにする必要がある。

これが，「プロセスの常駐終了」と呼ばれるものである。まず最初の問題は，デバッガで入力したプログラムをXファイルなりRファイルとして作成し，これを実行すれば解決する。既存の仕組みをそのまま利用するわけである。デバッガでXファイルを作成するのは非常に面倒だが，Rファイルなら簡単に作成できるので，この点は問題なかろう。次の問題だが，これはそのために特別に用意された方法でプログラムの実行を終了するようにすればいい。通常のプログラムは，

```
    _exit
```
としてその実行を終了するが，これを，
```
    move.w    #code,-(sp)
    move.l    #length,-(sp)
    -keeper
```
として終了させれば，Human68kはプログラムの実行後もそのメモリブロックを確保しておいてくれるのである。ここでcodeというのはプログラム終了時に返す終了コード，lengthというのはプログラム実行後も確保し続けるメモリブロックの長さである。

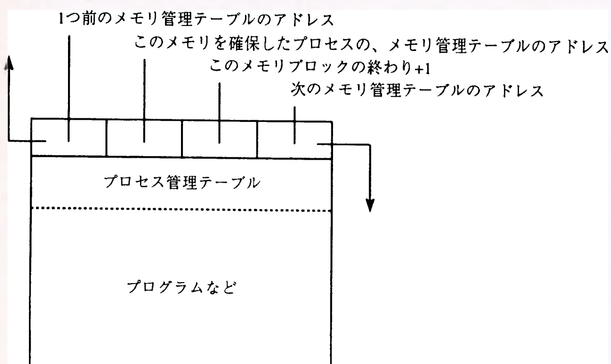
lengthについてもう少し補足しておこう。常駐して動作するプログラムは，

1) 常駐してさまざまな処理を行うプログラム
2) 1)をメモリに常駐させるためのプログラム

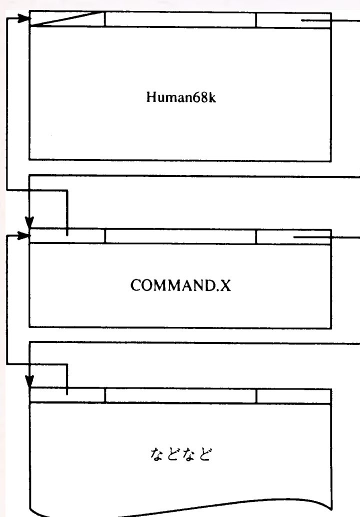
の2つの部分から構成される。しかしながら2)の部分は，プログラムが常駐してしまえば必要ない。そんなもののためにメモリを食われるのは面白くない，と世のプログラマは考えたのだな。で，本当に常駐しなければならない1)のサイズを指定すれば，「プログラム先頭から指定されたサイズ分」だけが常駐する仕様が考え出されたというわけである(図2)。この常駐するサイズは重要で，効率よく2)の部分を切り放すためには，常駐プログラムは図2-1のような構成にしておく必要がある。

図1　Human68kのメモリ管理の様子

1) Human68kが管理するメモリのブロック構造




2) このブロックが連なってメモリを構成する



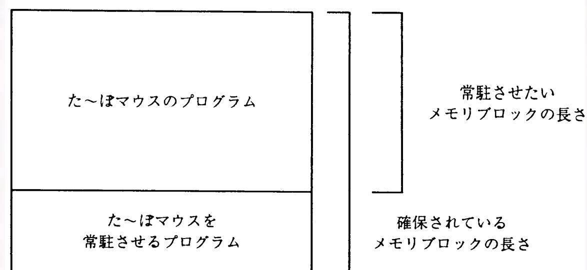

## ◆た〜ぼマウスを常駐させる

では，前回お届けしたた〜ぼマウスのプログラムを常駐プログラムに直した例をお目にかけよう。図3である。このプログラムをデバッガで入力し，200000H〜20008DHを，
```
    -w tmouse.r,200000 20008d
```
としてファイルに書き出せば，た〜ぼマウスのプログラムtmouse.rが完成する。もちろん，前回触れたように各自のROM内アドレスに合わせて，contとラベルが振ってある部分のジャンプ先は調整していただきたい。また，XVIでは割り込み処理プログラムの一部が図3とは異なっているので，その点も注意していただきたい。では，内容をざっと紹介しておこう。前半部分は前回お届けした割り込み処理プログラムとまったく同じなので内容は省略させていただく。今回の目玉はstartとラベルが振ってある行以降である。

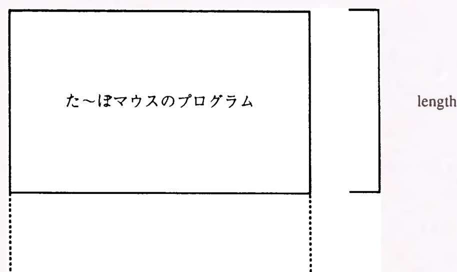
まず，Rファイルとはどのようなファイルなのかを復習しておこう。ご存じのように，アセンブラやデバッガで作成したプログラムは，通常決まったアドレスでしか動作することはできない。これは，
```
    move.w   #133,$200200
```
とか，
```
    jmp      $200100
```

図2　lengthの意味

1) た〜ぼマウスのプログラムのメモリ上の様子




2) 不必要な部分を切り放して常駐させる



ように，特定のアドレスを参照する命令が含まれているためである。Xファイルはこのようなプログラムがどのアドレスに読み込まれても動くように，読み込まれたアドレスに合わせてプログラムの一部を書き換えるような機構を備えている。この機構はアセンブラとリンカによって用意されるので，諸兄が心を煩わすほどのものではない。ただし，これをデバッガによって作成しようとすると，かなりの骨折りを覚悟する必要がある。

これに対してRファイルは，最初からどのアドレスに読み込まれても動作するように作成されたプログラムである。MC68000は，現在どのアドレスに入っている命令を実行しているのかを保持するレジスタPCをもっているので，これを使って「現在のPCから100バイト後ろにデータを読み込め」とか，「現在のPCから25バイト前にジャンプしろ」というようにプログラムを作成するのである。これなら，$200100_H$というように直接アドレスを指定することなくプログラムが作成できる。しかも，デバッガで入力したプログラムをファイルに保存するだけでRファイルが作成できる，まことに便利なものである。すべてをPCからの相対的なアドレスで記述するのはちょっと面倒だが，デバッガユーザーには見逃せないプログラム作成方法といえよう。Rファイルは先頭からプログラムが実行されるので，図３では先頭にstartへのBRA命令が書き込んである。常駐部分に先頭のBRA命令が残ってしまうが，４バイト程度ならあきらめてもよかろう。

さて，start以降のプログラムだが，最初にやっているのはSCCポートB割り込みの処理アドレスの設定である。前回は，

```
pea    get_ms_data
```

とやって，アドレスを直接スタックに積んでいたのだが，このget_ms_dataというアドレスは，プログラムが読み込まれるアドレスによって変化する。Rファイルにするためには，

```
pea    オフセット(pc)
```

としてアドレスを指定しなければならない。つまり，PCからオフセットバイト離れたところのアドレスをスタックに積め，というわけである。アセンブラなら，

```
pea    get_ms_data(pc)
```

と書けば，アセンブラが自動的にget_ms_dataまでのオフセットを計算してくれるのだが，デバッガにはこのような機能はないので，

－(PC－peaしたいアドレス)

というオフセットを自分で計算しなければならない。「4－$72」というのは，こうして求めたオフセットで，

$-(200072_H-200004_H)$

という計算を，簡略化して書き込んである。

なぜ「4-$70」ではなく「4-$72」なのかと疑問をもたれるかもしれない。それは，

```
00200070  pea  4-$72(pc)
```

という命令は，

```
00200070  487A  FF92
```

というコードになる。MC68000は，まず$487A_H$を読み込んでこの命令がPC相対のPEAだということを知るのだ

**図3　デバッガ版た〜ぼマウス常駐プログラム**

```
-z0=200000
-an .z0

          †_intvcs       equ     $ff25
          †_keeppr       equ     $ff31

          †       bra     start
00200000          bra     .z0+$70

          *
          *       差し替えプログラム
          *

          get_ms_data:
00200004          ori.w   #$0700,SR
00200008          movem.l D0-D1/A0-A1,-(A7)
0020000C          move.w  $00E98002,D0
00200012          movea.l $0000092C,A0
00200018          move.b  D0,(A0)+
0020001A          move.l  A0,$0000092C
00200020          subq.w  #1,$0000092A
          †       beq.s   check_x
00200026          beq.s   .z0+$36
00200028          movem.l (A7)+,D0-D1/A0-A1
0020002C          move.w  #$0038,$00E98000
00200034          rte

          check_x:
00200036          move.b  $931,d0
0020003C          cmpi.b  #$fd,d0
          †       blt.b   setnewx
00200040          blt.b   .z0+$48
00200042          cmpi.b  #3,d0
          †       ble.b   check_y
00200046          ble.b   .z0+$50
          setnewx:
00200048          add.b   d0,d0
0020004A          move.b  d0,$931

          check_y:
00200050          move.b  $932,d0
00200056          cmpi.b  #$fd,d0
          †       blt.b   setnewy
0020005A          blt.b   .z0+$62
0020005C          cmpi.b  #3,d0
          †       ble.b   cont
00200060          ble.b   .z0+$6a
          setnewy:
00200062          add.b   d0,d0
00200064          move.b  d0,$932

          cont:
0020006A          jmp     $ff1526   * ROMのプログラムへ戻す

          *
          *       Turbo Mouseを常駐終了させる
          *

          start:
          †       pea     get_ms_data(pc) * get_ms_dataを
00200070          pea     $4-$72(pc)
00200074          move.w  #$54,-(sp) * 割り込みベクトル54Hの
          †       dc.w    _intvcs
00200078          _intvcs              * 処理アドレスとして登録
0020007A          addq.l  #2,sp
0020007C          move.w  #$55,-(sp) * 割り込みベクトル55Hも同様
          †       dc.w    _intvcs
00200080          _intvcs              * 登録する
00200082          addq.l  #6,sp
00200084          clr.w   -(sp)        * 終了コードは0
00200086          move.l  #$70,-(sp) * サイズは70H
          †       dc.w    _keeppr      * 常駐
0020008C          _keeppr
```

† XVIユーザは、get_ms_data:の「ori.w……」の次の行に

```
bset.b  #5,$933
```

を補うこと。

また、これによってcheck_x以降の各ラベルのアドレスは

```
check_x          .z0+$3e
setnewx          .z0+$4e
……               ……
```

と8バイトずつ大きくなる。Bccのジャンプ先に注意されたい。

さらに、start:の先頭でget_ms_dataのアドレスをセットところもこれに合わせて

```
pea     $4-$7a(pc)
```

となり、常駐部分のサイズも

```
move.l  #$78,-(sp)
```

と変更する必要がある。

が，知ったときにはPCは487A$_H$に続くオフセットのアドレスを指してしまっている。つまり，PC＝200072$_H$になっているのである。このため，

　－(200070$_H$－200004$_H$)

ではなく，

　－(200072$_H$－200004$_H$)

としてオフセットを計算する必要がある。ちなみにFF92$_H$というのは，－6E$_H$をワード長の2の補数表現で表したものである。200072$_H$－6E$_H$＝200004$_H$となることを確認してみていただきたい。

SCCポートB割り込みの処理アドレスを変更したら，_keepprを使って常駐終了である。終了コードは通常0なので，ここでも0にしてある。そして常駐部分のサイズは，startの直前までの70$_H$バイトである。

図4　SCCポートB割り込みを元に戻すプログラム

```
-z0=200000
-an .z0

        †_exit          equ     $ff00
        †_intvcs        equ     $ff25

00200000    pea       $ff1502      * 各々のオリジナルベクタを書き込む
00200006    move.w    #$54,-(sp)   * 割り込みベクトル54Hの
        †   dc.w      _intvcs
0020000A    _intvcs                * 処理アドレスとして登録する
0020000C    addq.l    #2,sp
0020000E    move.w    #$55,-(sp)   * 割り込みベクトル55Hも同様に
        †   dc.w      _intvcs      * 登録する
00200012    _intvcs
00200014    addq.l    #6,sp
        †   dc.w      _exit        * 終了
00200016    _exit
```

図5　た～ぼマウスの常駐を解除する

```
E:\>tmouse          ←────   た～ぼマウスを常駐させる
E:\>process
X68k Process v2.00 Copyright 1989 SHARP/Hudson
--------------------------------------------------
開始   終了   長さ   モード   ファイル名
------ ------ ------ ------ ----------------------------
006800 073FFF 06D800 SUPER  Human.sys
074010 0740FF 0000F0 MALLOC
074110 07450F 000400 MALLOC
074520 07560F 0010F0 USER   A:\COMMAND.X
075620 07C96D 00734E MALLOC
07C980 07D39B 000A1C MALLOC
07D3B0 07D6AF 000300 MALLOC
07D6C0 07D8BF 000200 MALLOC
07DA40 37FFFF 3025C0 USER   C:\BIN\DOS\process.x

07D8D0 07DA2F 000160 KEEP   E:\tmouse.r    ←────  ここに常駐
E:\>vreset          ←──────────  割り込みベクタを元に戻して
E:\>db
X68k Debugger v2.10 Copyright 1987,88,89 SHARP/Hudson
-an 200000
00200000        pea $7d8d0  ┐
00200006        _mfree      │ メモリブロックを解放
00200008        _exit       ┘
0020000A        .
-g=200000
program terminated normally .
-q
E:\>process
X68k Process v2.00 Copyright 1989 SHARP/Hudson
--------------------------------------------------
開始   終了   長さ   モード   ファイル名
------ ------ ------ ------ ----------------------------
006800 073FFF 06D800 SUPER  Human.sys
074010 0740FF 0000F0 MALLOC
074110 07450F 000400 MALLOC
074520 07560F 0010F0 USER   A:\COMMAND.X
075620 07C96D 00734E MALLOC
07C980 07D39B 000A1C MALLOC
07D3B0 07D6AF 000300 MALLOC
07D6C0 07D8BF 000200 MALLOC
07D8D0 37FFFF 302730 USER   C:\BIN\DOS\process.x

            ←────   ほら、いなくなった
```

## ◆た～ぼマウスの常駐解除

常駐終了したた～ぼマウスを取り外すためには，2つのステップが必要となる。まず，変更したSCCポートB割り込みの処理アドレスを元に戻すこと。そして，Human68kが保持しているた～ぼマウスのためのメモリを解放することである。割り込み処理アドレスを元に戻す方法だが，これは図4のようなプログラムを作り，

　－w vreset.r,200000 200017

として登録すればOKである。vreset.rを実行すれば，tmouse.rが変更した割り込み処理アドレスは元のROM内ルーチンのアドレスに戻される。最初の行でスタックに積んでいるアドレスは，諸兄のマシンで確認していただきたい。マシン起動直後にアドレス150$_H$を参照し，そこに書き込んであるアドレスをここに書けばいい。

次にtmouse.rが常駐しているメモリブロックの解放だが，これにはDOSコールFF49$_H$を使用する。これは，

```
pea     アドレス
_mfree
addq.l  #4,sp
```

のように使えば，指定したアドレスのメモリブロックを解放するようになっている。指定するアドレスは，メモリ管理ポインタが入っている領域の次，つまり，メモリブロックの先頭アドレス＋10$_H$である。ちなみにこれは，プロセス管理テーブルのアドレスに相当するが，難しく考える必要はない。PROCESSコマンドを実行したときに表示される「開始アドレス」がそれに相当する。

図5をご覧いただきたい。これはtmouse.rが常駐しているメモリを解放しているところである。実に簡単に解放できるのがおわかりいただけるかと思う。デバッガで入力しているプログラムでは，本来スタックの補正をしなければならないが，どうせ終了するのだからと省略してある。実害はないが，ま，悪いお手本といえよう。

## ◆今年のクリスマスプレゼント

去年のクリスマスには簡易カリグラフィソフトともいうべきものをお届けした。今年のプレゼントは，た～ぼマウスのプログラムである。吾輩としては，ハードウェアやOSにべったりとくっついた図3，4，5の方法で吾輩のマウスを高速化していただくのが嬉しいのだが，面白くはあるが面倒だという諸兄も少なくあるまい。とりわけ，うちの御仁のような物臭太郎にとっては，面倒だというのは致命的ですらある。

御仁は前回紹介したデバッガ用のプログラムを作成したのちすぐさま，た～ぼマウスの常駐，常駐解除プログ

ラムを作成し，AUTOEXEC.BATで自動実行するようにしてしまった。今回は御仁の作ったこのプログラムをもとに，XVIとそれ以前のマシンを自動判別するよう改造したものをお届けしよう。

自動判別の方法に入る前に，た〜ぼマウスのプログラムがどのように動いているのかをここで復習しておきたいと思う。図6をご覧いただきたい。もし，SCCポートB割り込みの処理アドレスが変更されていなければ，割り込みは図の左側のルートを通って処理される。た〜ぼマウスが常駐して割り込み処理アドレスが書き換えられると，図の右側のルートが利用される。ただし右側のルートも，処理が終了すれば左側のルートに合流する点に注意していただきたい（図3のアドレス20006A$_H$を参照）。合流地点は，

```
    lea     $930,A1
```

という命令のある行である。

この合流地点は，XVI以前のマシンでは，オリジナルの割り込み処理プログラムの先頭アドレス＋24$_H$になっている。割り込み処理プログラムの先頭アドレスはマシンによって異なっているのだが，そこから24$_H$あとには必ずこの命令が待っているのである。XVIの場合は，オリジナルの処理アドレス＋2E$_H$が合流地点となる。この違いを利用すれば，XVIとそれ以前のマシンを区別することが可能となる。

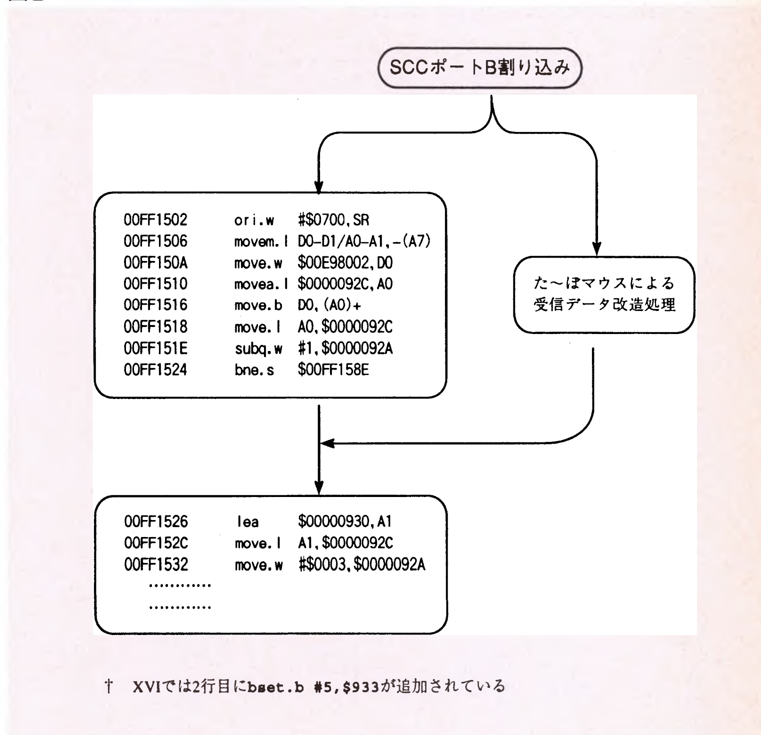
もちろんこれは，一般的な機種の判定方法とは大きく異なる判定方法であることをお断りしておく。一般には，ROMのバージョンをチェックしてマシンを判定するのが普通だ。ここでその方法を使用しなかったのは，図6のようにオリジナルの割り込み処理プログラムの中に飛び込むという特異な処理を行っているためである。一般には公開されていないROM内プログラムの，しかもその途中に飛び込むので，ROMのバージョンだけを信用するというのはいささか心許ない。XVIシリーズでは，今後ROMのバージョンは同じであるにもかかわらず，この微妙なアドレスが変更される可能性がなきにしもあらず，という危惧があったからなのである。

プログラムは図7である。残念ながらこのプログラムはデバッガで作成できるようにはなっていない。少々長いので，デバッガ用にRファイル形式に書き直すのが面倒だったためである。あぁ，なんたること。吾輩にも御仁の不精癖がうつってきたのかもしれない。ま，本格的な割り込み処理プログラムはアセンブラで作成するのが筋，ということで勘弁していただきたい。

まず冒頭であるが，ここには「tmouse.x」という文字列がセットしてある。Xファイルはプログラムの実行開始アドレスを自由に設定できるようになっているので，冒頭にこのように文字を書き込んでも差し支えない。この文字列は，「このメモリブロックはた〜ぼマウスが使用している」ということを判定しやすくするために入れてある。図8をご覧いただきたい。これは図7のプログラムをアセンブル，リンクしてできるtmouse.xを実行したときのメモリの様子である。メモリ管理ポインタ16バイトとプロセス管理ポインタ240バイト，合計100$_H$バイトのあとにプログラムが格納されるが，この冒頭に「tmouse.x」という文字列が書き込まれることになる。この文字列はた〜ぼマウスの常駐を解除するときに参照するので，あとで改めて説明することにしよう。

続いて図7のリストを眺めていくと，tmouseXVIとtmouseの2つのラベルがあるのを確認していただけるだろう。マシンがXVIなら前者を，それ以前の機種ならば後者を，SCCポートB割り込みの処理プログラムのエントリとして登録する。違いは，2行目に「bset.b……」

**図6　た〜ぼマウスの処理動作**




† XVIでは2行目にbset.b #5,$933が追加されている

**図7　た〜ぼマウスのプログラム**

```

*    た～ぼマウスのプログラム

_exit           equ     $ff00
_keeppr         equ     $ff31
_mfree          equ     $ff49
_intvcs         equ     $ff25
_intvcg         equ     $ff35

progname:
        dc.b    'tmouse.x'
*
*           差し替えプログラム
*
tmouseXVI:                                         * XVI用エントリ
        ori.w   #$0700,SR
        bset.b  #5,$933
        bra     tmouse1
tmouse:                                            * それ以前用エントリ
        ori.w   #$0700,SR
tmouse1:
        movem.l D0-D1/A0-A1,-(A7)
        move.w  $00E98002,D0
        movea.l $0000092C,A0
        move.b  D0,(A0)+
        move.l  A0,$0000092C
        subq.w  #1,$0000092A
        beq.s   check_x
        movem.l (A7)+,D0-D1/A0-A1
        move.w  #$0038,$00E98000
        rte

check_x:
        move.b  $931,d0
        cmpi.b  #$fd,d0
        bcc.b   check_y
        cmpi.b  #3,d0
        bls.b   check_y
        add.b   d0,d0
        move.b  d0,$931
```

```
check_y:
        move.b  $932,d0
        cmpi.b  #$fd,d0
        bcc.b   cont
        cmpi.b  #3,d0
        bls.b   cont
        add.b   d0,d0
        move.b  d0,$932
cont:
        movea.l jmpadrs,a0
        jmp     (a0)

jmpadrs:
        dc.l    0
orgvect:
        dc.l    0

*********************************

start:
        tst.b   (a2)                * パラメータチェック
        beq     vctcheck            * パラメータなし

        lea     usage,a1            * 使用法アドレスをセット
        cmpi.b  #2,(a2)
        bne     error
        cmpi.b  #'r',2(a2)
        beq     start1              * パラメータは-r
        cmpi.b  #'R',2(a2)
        bne     error               * パラメータが-Rでない
start1:
        bsr     svcheck             * SCCポートBベクタチェック
        bcs     namecheck           * 書き替えられている
        lea     nochange,a1
        bra     error

namecheck:                          * プロセス名をチェック
        movea.l (a0),a2
        lea     $100(a2),a2
        lea     progname,a3
        moveq   #8-1,d0
namecheck1:
        cmpm.b  (a2)+,(a3)+
        bne     namecheck2          * プロセス名が違う
        dbra    d0,namecheck1
        bra     remove
namecheck2:
        lea     impossible,a1
        movea.l (a0),a0
        beq     error               * プロセスが見つからない
        bra     namecheck

remove:                             * 常駐解除
        lea     removed,a1
        moveq   #$21,d0             * _b_print
        trap    #15
        movea.l (a0),a1             * SCCチャンネルB割り込みを元に戻す
        move.l  $100+orgvect-progname(a1),d0
        move.l  d0,-(sp)
        move.w  #$54,-(sp)
        dc.w    _intvcs
        move.w  #$55,(sp)
        dc.w    _intvcs
        addq.l  #6,sp

        movea.l (a0),a0
        lea     $10(a0),a0
        move.l  a0,-(sp)
        dc.w    _mfree
        addq.l  #4,sp
        dc.w    _exit

vctcheck:
        bsr     svcheck             * SCCポートBベクタチェック
        bcc     vercheck            * 書き替えられていない
        lea     changed,a1
        bra     error               * ベクタが書き替えられている

vercheck:                           * ROMのバージョンチェック
        move.w  #$54,-(sp)          * SCCポートBベクタを取り出す
        dc.w    _intvcg
        moveq   #$24,d1             * XVI以前のマシンのオフセット
        add.l   d1,d0
        movea.l d0,a1               * ベクタ+$24を調べる
        moveq   #$83,d0             * _b_wpeek
        trap    #15
        lea     tmouse,a0           * 新割り込みベクタ
        cmpi.w  #$43f9,d0           * lea ……,a1 か?
        beq     stay
        lea     8(a1),a1            * ベクタ+$2eを調べる
        moveq   #$83,d0             * _b_wpeek
        trap    #15
        lea     badver,a1
        cmpi.w  #$43f9,d0           * lea ……,a1 か?
        bne     error
        moveq   #$2e,d1             * マシンはXVI
        lea     tmouseXVI,a0        * 新割り込みベクタ

stay:                               * ターボマウス常駐
        lea     stayed,a1
        moveq   #$21,d0             * _b_print
        trap    #15
        move.l  a0,-(sp)            * 新ベクタを
        move.w  #$54,-(sp)          * SCCポートB割り込みにセット
        dc.w    _intvcs
        move.w  #$55,(sp)
        dc.w    _intvcs
        addq.l  #6,sp
```

が入っているかどうかだけである。

そのあとは，contというラベルのところまで諸兄がご存じのプログラムと同様である。さて，contだが，これまではROM内の処理プログラムの途中に直接ジャンプしていた。しかし，全機種対応となるとそうは簡単にいかない。ここでは，jmpadrsというワークに保存してあるアドレスを取り出してジャンプするように変更してある。jmpadrsには，常駐時に「オリジナルの処理アドレス＋$24_H$，あるいは＋$2E_H$」が機種に応じてセットされる。また，jmpadrsの次のorgvectというワークは，オリジナルの処理アドレスを保存するためのものである。

ここまでが，メモリに常駐するプログラムである。これ以降はた〜ぼマウスを常駐させるプログラム，および常駐を解除するプログラムとなる。順に見ていこう。

## ◆起動時のパラメータチェック

プログラムが起動すると，コマンドラインのパラメータ文字列を収めたアドレスがA2.Lにセットされる。パラメータ文字列は，「文字列の長さ」＋「文字列本体」というフォーマットになっている。したがって，もしA2.Lにセットされているアドレスに0が入っていれば，パラメータ文字列はない，と判定できることになる。

起動されたた〜ぼマウスのプログラムは，

1)　パラメータがあるかどうか
2)　もしあれば，それが2文字かどうか
3)　もし2文字なら，それが－r，－Rかどうか

を順にチェックする。そして，パラメータがなければた〜ぼマウスの常駐処理を，パラメータが－rか－Rなら(1文字目はチェックしていないので，/rか/Rでもかまわない)常駐解除処理を行うようになっている。

## ◆常駐処理

常駐処理を行う場合には，まずSCCポートB割り込みが書き換えられていないかどうかをチェックする。これが書き換えられていれば，すでにた〜ぼマウスは常駐しているものとして処理を中断する。

書き換えられていなければ，冒頭に述べたマシンのチェックを開始する。vercheckとラベルを振った部分である。SCCポートB割り込みの処理アドレスを取り出し，これに$24_H$を加えたアドレスに「lea ……,a1」という命令を意味する$43F9_H$というデータが書き込まれているかどうかを調べるのである。もしそうならstayにジャンプして常駐処理を行う。そうでなければ＋$2E_H$のアドレスをチェックし，XVIかどうかの判定を行う。いずれでもなければ，新バージョンだということで，た〜ぼマウスはメッセージを表示して終了するようになっている。

常駐処理を行うstayでは，SCCポートB割り込みの変更，orgvectの保存，jmpadrsの設定という一連の処理を行う。プログラムを見ただけでは混乱なさるかもしれないので補足しておく。_intvcsは変更前に設定されていたアドレスをD0.Lに返すようになっている。これをそのままorgvectに保存し，機種によるオフセット($24_H$

か$2E_H$)を加えてjmpadrsを設定しているのがこのプログラムである。最後に，終了コードと常駐サイズを指定して，た〜ぼマウスは常駐終了する。

## ◆常駐解除処理

常駐を解除するときにも，まずSCCポートB割り込みの処理アドレスが変更されているかどうかをチェックする。変更されていれば，常駐しているた〜ぼマウスが使用しているメモリブロックの検索である。

た〜ぼマウスが常駐しているメモリブロックを判定するには，メモリブロックの先頭に$100_H$バイト加えたアドレスから8バイトをチェックし，それが「tmouse.x」かどうかを調べればいい。先の伏線がここにきて役に立つというわけである。

図8に補足しておいたのだが，プログラムを実行したときA0.Lは自分自身のメモリブロックの先頭アドレスを保持している。ここからイモヅル式にメモリブロックのリンケージを遡りながら「メモリブロックの先頭+$100_H$」をチェックしていくことで，すでに常駐しているた〜ぼマウスのプログラムが使用しているメモリブロックを突き止めることができる。

メモリブロックを突き止めてしまえばあとは簡単。保存されているorgvectを取り出し，それをSCCポートB割り込みの処理アドレスとして再登録。メモリブロックを解放すればいい。メモリブロックを解放するときには，プロセス管理テーブルのアドレス，つまり，メモリブロックの先頭+$10_H$を指定する仕様になっているのをお忘れなく。図7では，removeとラベルを振った部分がこの処理を行っている。

## ◆た〜ぼマウスが速すぎる

これまで吾輩のマウスに慣れ親しんできた諸兄は，た〜ぼマウスはいささか速すぎるとお感じになるかもしれない。前回も解説したように，このプログラムはマウスが送ってきた生のデータをこっそり2倍にすることでx，y方向の移動量を拡大している。プログラムでいうなら，check_x，check_yとラベルをつけたところである。ここで生のデータを1.5倍にする程度にとどめれば，マウスカーソルの動きをもう少し穏やかにすることができる。

変更方法は，

```
    add.b     d0,d0      * これで2倍
    move.b  d0,$931
```

とする代わりに，

```
    asr.b   #1,d0        * D0.Bを半分に
    add.b   d0,$931
```

となされればいい。また，$-3 \leqq$ 移動量 $\leqq 3$ のときにはデータを2倍しないようになっているが，この範囲をもう少し拡大なさってもいいかもしれない。いずれにせよ，た〜ぼマウスは諸兄のものである。ご自分の使いやすいように微調整を加えて使っていただければ，全機種対応にした吾輩の苦労も報われようというものだ。活用していただきたい。

```
          move.l  d0,orgvect       * オリジナルベクタを保存
          add.l   d1,d0            * オリジナルベクタ+オフセット
          move.l  d0,jmpadrs       * jmpadrsに保存

          clr.w   -(sp)
          move.l  #start-progname,-(sp)
          dc.w    _keeppr

svcheck:                           * SCCポートBベクタチェック
          lea     $151,a1
          moveq   #$82,d0          * _b_bpeek
          trap    #15
          cmpi.b  #$f0,d0
          rts

error:
          moveq   #$21,d0          * _b_print
          trap    #15
          dc.w    _exit

usage:
          dc.b    $0d,$0a
          dc.b    'Turbo Mouse',$0d,$0a
          dc.b    'usage : tmouse [opt]',$0d,$0a,$0d,$0a
          dc.b    'opt  : 機能',$0d,$0a
          dc.b    'なし : ターボマウス常駐',$0d,$0a
          dc.b    '-r   : 常駐解除',$0d,$0a
          dc.b    $0d,$0a,0

nochange:
          dc.b    $0d,$0a
          dc.b    'SCCポートB割り込みは変更されていません'
          dc.b    $0d,$0a,$0d,$0a,0

changed:
          dc.b    $0d,$0a
          dc.b    'SCCポートB割り込みが変更されています',$0d,$0a
          dc.b    '常駐せずに終了します'
          dc.b    $0d,$0a,$0d,$0a,0

impossible:
          dc.b    $0d,$0a
          dc.b    'ターボマウスは常駐していません'
          dc.b    $0d,$0a,$0d,$0a,0

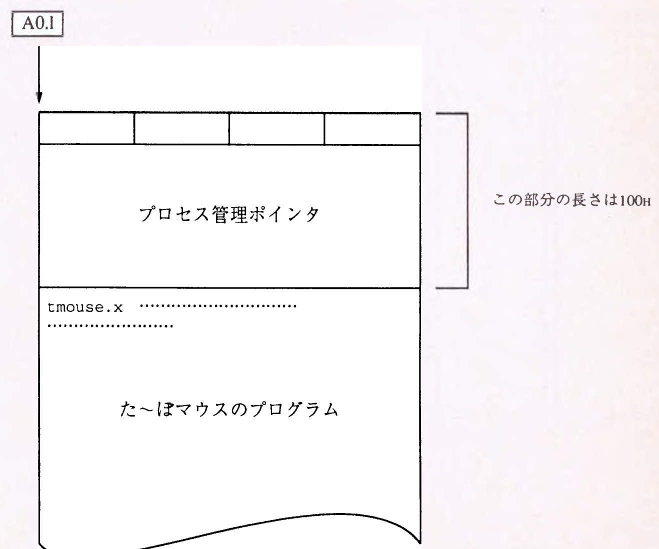
stayed:
          dc.b    $0d,$0a
          dc.b    'ターボマウスが常駐しました'
          dc.b    $0d,$0a,$0d,$0a,0

removed:
          dc.b    $0d,$0a
          dc.b    'ターボマウスの常駐を解除しました'
          dc.b    $0d,$0a,$0d,$0a,0

badver:
          dc.b    $0d,$0a
          dc.b    '残念ながらターボマウスは',$0d,$0a
          dc.b    'このマシンには対応していません'
          dc.b    $0d,$0a,$0d,$0a,0

end:
          .end    start
```

図8　た〜ぼマウス実行時のメモリイメージ



X68000用CARDDRV対応カードゲーム

# GOLF

Takayama Tadanobu 高山 忠信



タイトルのとおり，ゴルフのイメージで手札をなるべくなくさず，場札を効率よくなくしていくこのゲーム。あまり難しくなく，誰でも手軽に簡単に楽しめるものに仕上がっています。ところで，このゲームにも幻のホールインワンは存在するのでしょうか。

今回発表するカードゲームは「GOLF」。まずクラブ，次にショットパワーを選び……。そういったゴルフではありません。これはカードゲームのゴルフです。

台札に続くカードを場札から重ね，最終的に場札にあるすべてのカードをなくさなくてはなりません。

本物のゴルフのような成績は出ませんが，一発クリアのホールインワンを目指して遊んでみてください。

## 入力方法

このゲームを実行するためには，CARDDRV.XとトランプデータTR.DAT，CARD2.FNC（CARD.FNCでも可）が必要になります。CARDDRV.Xはコマンドライン上から，

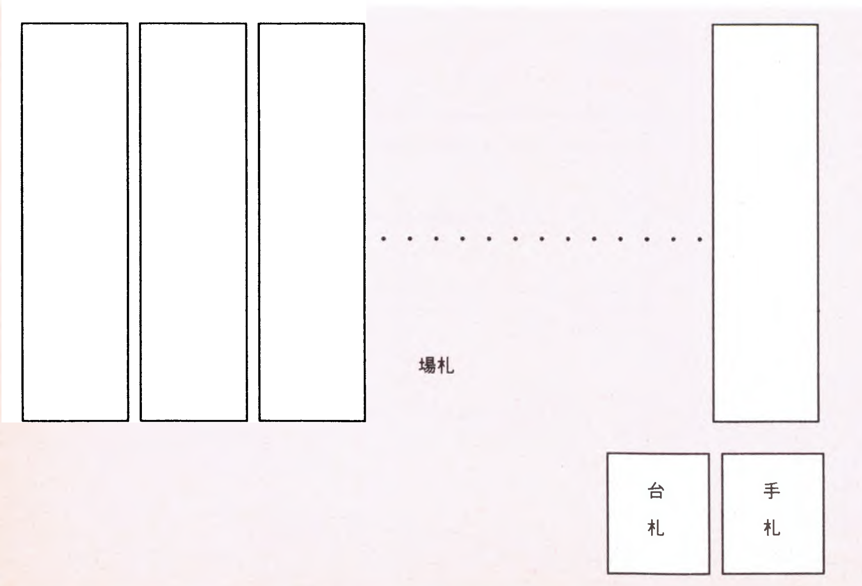
```
CARDDRV TR.DAT
```

として組み込み，CARD.FNCはX-BASICのコンフィグレーションファイルに，

```
FUNC = CARD2
```

と追加登録しておきましょう。

図1




プログラムリストは，X-BASICで記述されていますので，BASICを立ち上げてそのまま打ち込んでもらえば結構です。

コンパイルについては，なんの問題もなく行われますので，安心してコンパイルしてください。まあ，たいしたことはやっていないので，速度的にはBASICインタプリタでも問題ありませんけどね。

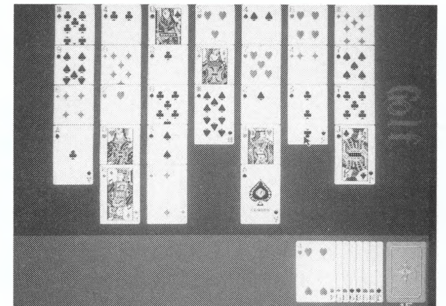
## 遊び方

ゲームを始めると，まず図1のような画面になります。

画面には，横7列×縦5列の場札が表にして並べられます。そして，画面右下に台札として，1枚だけ表になったカードが置かれ，残りの16枚が手札として台札の横に裏返しに置かれます。

ゲームの目的は，35枚の場札すべてをなくすことにあります。プレイヤーは，台札に続くカードを次々と場札から重ねなくてはなりません。

どうすればいいのかというと，下のほうに置かれた台札と数が続いたものを場札の中から選んでください。その選んだ札が自動的に次の台札となります。



台札に続けられる数は，台札の数字よりひとつ多いか，ひとつ少ないものです。なお，KからA，AからKへと続けることができませんので注意してください。

そして，台札に続く場札がなくなって行き詰まった場合，伏せてある手札をめくってください。これによって新しい台札が現れます。

手札をすべてめくり終わるとゲームオーバー。手札がなくなるまでに全部の場札を移動させることができれば，コングラチュレーションとなります。

## 終わりに

ゲーム自体は，結構成功率が高いでしょう。うまくいけば数回トライすればゲームを終わらせることができると思います。

プログラム自体もシンプルなもので，長さも適当です。ぜひ皆さんのカードゲームライブラリに加えてください。難しすぎてストレスが溜まるようなこともなく，気軽に遊べますから。

まだまだ，作品のストックはありますので，また，誌面でお会いすることがあるかもしれません。楽しみにしていてくださいね。

＜参考文献＞
「トランプの遊び方」有紀書房，桐山雅光著

## リスト1

```
10 /*
20 /* Golf
30 /* Programmed by T.Takayama
35 /* '91.12.29(Sun.)-'91.12.30(Mon.)
40 /*
50 int mx,my,bl,br
60 char chk,count,fnd,fnd_c
70 dim char card(51)
80 prep()
90 repeat
100   init()
110   while chk<35 and count<53
120     man()
130   endwhile
140 until replay()
150 screen 1,1,1,1
160 mouse(0)
170 end
180 /*   1ゲーム毎の初期化
190 func init()
200   int i,k,s,t
210   mouse(2)
220   for i=0 to 51
230     card(i)=i+1
240   next
250   for i=0 to 99
260     s=rnd()*52:t=rnd()*52
270     k=card(s):card(s)=card(t):card(t)=k
280   next
290   wipe()
300   c_put(461,395,0)
310   for i=0 to 34
320     w_ba(i)
330   next
340   chk=0
350   fnd_c=0
360   count=35
370   get_st()
380   mouse(1)
390 endfunc
400 /*   人間の処理
410 func man()
420   char pp
430   mson()
440   pp=select()
450   switch pp
460     case 35:get_st() :break
470     case 36:count=53 :break
480     case 37:m_play(2):break
490     default:move(pp)
500   endswitch
510   msoff()
520 endfunc
530 /*   プレイヤーの指すカード
540 func select()
550   int i,x,y=4,k,r
560   if (62<mx and mx<447) and (14<my and my<367) then {
570     x=(mx-63)¥56
580     for i=0 to 4
590       if card(x*5+4-i)=0 then y=3-i
600     next
610     if -1<y then {
620      if (y*64+14)<my and my<y*64+111 then {
630      r=x*5+y
640      if abs(number(card(r))-number(fnd))=1 then return(r)
650      }
660     }
670   }
680   if 460<mx and mx<509 then {
690     if 394<my and my<491 then {
700       if count=52 then {
710         if 428<my and my<455 then return(36)
720       } else return(35)
730     }
740   }
750   return(37)
760 endfunc
770 /*   場札を台札に移す
780 func move(a;int)
790   p_fnd(card(a))
800   card(a)=0
810   w_ba(a)
820   if a mod 5<>0 then w_ba(a-1)
830   chk=chk+1
840   m_play(1)
850 endfunc
860 /*   場札を描く
870 func w_ba(a;int)
880   int x,bx,by,y,y2
890   x=(a¥5)*56+63
900   by=a mod 5
910   y=by*64+15
920   if 0<card(a) then {
930     line(x,y,x+46,y,1)
940     c_put(x,y+1,card(a))
950   } else {
960     if by=0 then y2=0 else y2=32
970     fill(x,y+y2,x+47,y+96,0)
980   }
990 endfunc
1000 /*   手札をめくる
1010 func get_st()
1020   str n
1030   p_fnd(card(count))
1040   count=count+1
1050   fill(477,491,493,507,0)
1060   if count<52 then {
1070     n=itoa(52-count)
1080     symbol(487-len(n)*5,491,n,1,1,1,15,0)
1090   } else {
1100     fill(461,395,508,490,0)
1110   }
1120   m_play(1)
1130 endfunc
1140 /*   台札に加える
1150 func p_fnd(a;int)
1160   int x
1170   fnd=a
1180   x=410-fnd_c*8
1190   c_put(x,395,fnd)
1200   fnd_c=fnd_c+1
1210 endfunc
1220 /*   カードの(数字-1)を返す
1230 func number(a;int)
1240   return((a-1) mod 13)
1250 endfunc
1260 /*   マウスが押されるまで待つ
1270 func mson()
1280   repeat
1290     msstat(mx,my,bl,br)
1300   until bl or br
1310   mspos(mx,my)
1320 endfunc
1330 /*   マウスが離されるまで待つ
1340 func msoff()
1350   repeat
1360     msstat(mx,my,bl,br)
1370   until bl+br=0
1380 endfunc
1390 /*   リプレイ?
1400 func replay()
1410   str n
1420   if chk=35 then demo()
1430   apage(0)
1440   fill(153,210,357,234,6)
1450   if chk=35 then {
1460     symbol(154,211,"Congratulations!!",1,1,2,10,0)
1470     symbol(153,210,"Congratulations!!",1,1,2, 9,0)
1480   } else {
1490     n=itoa(35-chk)
1500     symbol(181-len(n)*6,210,n+"枚 残りました",1,1,2,15,0)
1510   }
1520   vpage(15)
1530   setmspos(235,291)
1540   msarea(215,282,295,300)
1550   mson()
1560   msarea(0,0,511,511)
1570   vpage(14)
1580   apage(2)
1590   return(259<mx)
1600 endfunc
1610 /*   クリア デモ
1620 func demo()
1630   mouse(2)
1640   m_tempo(120)
1650   m_play(3,4)
1660   repeat:until m_stat(3)=0
1670   m_play(5,6)
1680   repeat
1690     home(2,rnd()*2,rnd()*2)
1700     home(3,rnd()*2,rnd()*2)
1710   until m_stat(5)=0
1720   home(2,0,0)
1730   home(3,0,0)
1740   mouse(1)
1750 endfunc
1760 /*   準備
1770 func prep()
1780   int i
1790   str a,b
1800   randomize(val(mid$(time$,4,2)+right$(time$,2)))
1810   screen 1,1,1,1
1820   mouse(0):mouse(4)
1830   vpage(0)
1840   console ,,0
1850   palet(1,0)
1860   palet(8,rgb(5,18,7))
1870   palet(9,rgb(7,22,9))
1880   apage(3)
1890   fill(0,0,511,384,8)
1900   symbol(502,146,"Golf",2,2,2,10,3)
1910   symbol(506,142,"Golf",2,2,2,9,3)
1920   fill(0,385,511,511,9)
1930   fill(461,429,506,454,10)
1940   symbol(466,430,"END",1,1,2,13,0)
1950   apage(0)
1960   box(137,198,373,312,13)
1970   fill(138,199,372,311,6)
1980   box(139,200,371,310,13)
1990   symbol(195,243,"Try again?",1,1,2,5,0)
2000   fill(215,282,255,300,2)
2010   fill(263,282,295,300,2)
2020   symbol(223,284,"Yes   No",1,1,1,11,0)
2030   fill(461,395,508,507,9)
2040   vpage(14)
2050   apage(2)
2060   for i=1 to 6
2070     m_alloc(i,100):m_assign(i,i)
2080   next
2090   m_trk(1,"q3@45v11t200o2c4")
2100   m_trk(2,"q8@15v13t100o3c4")
2110   a="o7q8@65@v115a4&@v116a4.&@v118a2&@v95a4"
2120   b="o218@50ag+ag+ag.a.r4g4g+"
2130   m_trk(3,"p1"+a)
2140   m_trk(4,"r16p0"+a)
2150   m_trk(5,"r8"+b)
2160   m_trk(6,b)
2170 endfunc
```

## 製品紹介

X68000用CD-ROMの活用

# SX広辞苑

Kioi Makoto 紀尾井 誠

**計測技研から発売されていたCD-ROMドライブ用のアプリケーションとしてCD-ROM広辞苑用の検索ソフトが発売されます。SX-WINDOW上で動作して，語句の詳細な情報を表示できます。マルチウィンドウ環境とマッチした強力なツールです。**

想像してみてください。

マルチウィンドウの操作環境，ワープロ上で文書を編集しつつ，用語の詳しいデータがほしいといったとき，その単語をマウスでちょいとつまんでぽいっとやるとデータベースから詳しい解説や図版，音声データが出力される……。

＊　＊　＊

国語辞典というのは文章を書くあらゆる人々にとって強力な味方となります。CD-ROM広辞苑はこんな環境の先駆けとして発売されました。20万語に及ぶ見出し語に対して，語義や語源，用例などの解説が収録されています。なにしろ中身は定評のある広辞苑そのものです。比較的早期に作られたCD-ROMソフトであるにもかかわらず，文字情報のみならず約2000点の図版や音声などマルチメディアを意識した構成になっていることも注目すべきでしょう。

このようなデスクトップのデータベース検索システムにとって，マルチタスク，マルチウィンドウの環境は重要な意味を持ってきます。SX広辞苑はいくつでも同時に起動することができます。複数の項目を同時に参照しながらの文書作成などが可能です。ほかのアプリケーションとのデータのやり取りなどを考えると，この検索ソフトがSX-WINDOW用として開発されたのは正解といえるでしょう。

### 使用上の注意

CD-ROMソフトを扱う場合，著作権に関する問題が微妙な様相を呈してきます。安心して使えるものか否かは必ず確認する必要があります。

たとえば，マイクロソフトのMultimediaシリーズなどではCD-ROMソフトの使用を同時に1台のマシンでしか許容していません。ソフトの使用に関する定義も明確で，メモリや外部記憶上にロードされている状態をもって使用しているとみなされます。こうなると，キャッシュつきのCD-ROMドライブとかディスクキャッシュソフト，さらにはハードディスクを使ってCD-ROMを高速化するソフトなどの存在が微妙になってくるのですが，アメリカでもまだ問題視されていないので，ここでは深く追及しないことにしましょう。

で，問題のCD-ROM広辞苑ですが，このソフトウェアでは個人使用に関する限りの複製までは認められていますので，安心して使用することができます。将来的にハードディスクでキャッシングを行うソフトなどが現れた場合でも心配はありません。

もうひとつの焦点となる引用に関しては出典を明記するように指導されているようです。引用というのもソフトウェアの部分複製にあたりますから一部のCD-ROMソフトでは注意が必要です。まれに，参照はできても引用はできそうにない変な規定のものもありますので。

たとえば，PC-9801用の一太郎から広辞苑を検索するシステムではカット&ペーストを行うと自動的に出典明記の文が追加されるという親切設計になっています。SX広辞苑ではさすがにそこまではやっていません。各自で出典明記を行ってください。

### SX広辞苑を使う

お待たせしました。では，SX広辞苑の使い勝手を見てみることにしましょう。

まず岩波書店から発売されている12cm版のCD-ROM広辞苑第三版を用意します。FM TOWNS版のCD-ROM広辞苑がもっとも手軽に使えるものとなっています。データの内容は同じなのでPC-9801版やMac広辞苑でもいいのですが，なぜか値段は違います。

CD-ROMをアクセスできる状態にしてSX-WINDOWを立ち上げ，SX広辞苑.Xを起動します。起動時の画面では広辞苑使用の際の注意点が表示されます。

SX広辞苑の使い道というのは，もちろん語句に関するデータの検索です。SX広辞苑では数種類の検索方法を提供しています。

気になる検索速度ですが，せいぜい数秒といったところでかなり高速です。30Mバイトに及ぶ文書データも相当なものですが，これに対するインデックスデータは118Mバイトが用意されています。ドライブ自体の高速さも生きているようです。

**●入力語検索**

これはもっとも基本になる使い方です。

検索文字列として読みがな，または検索語そのものが使用できます。読みがなの場合は同音意義語もすべて検索され，漢字の文字列などで直接指定したときには，その語句だけの解説を出力します。




CD-ROM広辞苑




SX広辞苑

検索語入力はメニューの「検索」の部分，またはショートカットキー(OPT.1+F)で疑似ダイアログを呼び出して行います。ウィンドウ内でカーソルが点滅しているのになぜ疑似ダイアログを呼び出さなければならないのかよくわかりませんが，単語の入力以外の操作はすべてマウスで行われます。

ここで単語の読みか単語そのものを入力すると，その単語を見出しとした解説文をウィンドウの下側に出力します。上の段には見出し語だけ抽出されて並びます。見出し語の候補がたくさんあるときは希望の見出し語をクリックするとすぐに対応する解説が表示されます。

疑似ダイアログの部分で右クリックを行うことにより過去の検索データ30件分のヒストリーがメニューになって現れます。同じデータを使うときには便利な機能ですが，このため文中からコピーした文字列を検索文字列としてペーストするといったことはできなくなっています。

**●入力語検索2/●画面語検索**

また，出力されたデータ中の文字列を指定して検索させることも可能です。

解説文中で使用されている文字列を使って検索を行うものには2種類があります。ひとつはその単語そのものを見出し語として検索する入力語検索2，そして，その単語を解説文中で参照している単語を検索する画面語検索です。残念ながらあまり適切な呼び名とは思えません。

ウィンドウ中の解説文から探したい文字列をマウスで指定します。あとはメニューまたはショートカットキーで入力語検索2を指定します。

**●条件検索**

解説文中で使用されている単語を条件とした検索です。キーワードとして複数の単語が指定できます。試しに，「たこ」「いか」と入れてみたところ「おうむがい」などの3件が検索されました。ひとつずつ処理されていくので条件が増えても気になるほど遅くはなりません。

**●前方/後方一致検索**

これは検索語指定の方法なのですが，入力する文字列にワイルドカード「*」を指定して検索を行うことができるのです。

*とんぼ

とすると，「あかとんぼ」や「しおからとんぼ」などが出てくるわけです。語の前後に1個だけ使えるという，Human68kと同じような制限があります。

*　　*　　*

やや，問題なのは，条件検索や画面語検索では解説文にその単語が使用されているものすべてを取り出してくれないことです。広辞苑自体の問題か検索ソフト側の問題かはっきりしません。

また，ウィンドウが上下に分割されているのですが，入力語検索以外では上のウィンドウに項目を並べてくれないのです。図版があるかどうかなどは上のウィンドウでしか知らされませんからちょっと不便です。

## グラフィック表示

CD-ROM広辞苑には約2000点の図版と鳥の鳴き声60種，そして色を表す語のためのカラー表示といった補助的な項目が用意されています。これらがすべてSX広辞苑でサポートされています。

表示された図版や色見本はクリップボードへ転送してグラフィックエディタで使用することができます。

広辞苑本体はOPT.1キーとの併用で非アクティブ状態での動作もできるのですが(要するにほかのウィンドウの下でも動く)，図版はこれに対応しません。エディタと併用するときは常にOPT.1を押して操作しているので(操作性が格段によい)，図版を扱うと多少うっとうしいことになります。

また，本来のCD-ROM広辞苑についていたはずの図版一覧機能がなくなっているのも少しさみしい気がします。

## 今後の課題

マルチウィンドウ環境では複数の広辞苑が立ち上がります。語句の内容を対比してみたいことはよくあります。すると現在表示されているものを破棄するのは嫌ですから次の広辞苑を立ち上げます。しかし，前のウィンドウ内で表示された語句を指定して検索するといったことはできません。カット&ペーストが禁止されているのでいちいち手で入力しなければならないのです。

広辞苑には日本語界で広く認められた辞典であるというメリットがあります。逆に広辞苑であることのデメリットもあります。それは収録されている単語がイマイチ現代的でないことでしょう。文系の学生が日本史関係のレポートを書くといった場合には非常に心強いのですが……。

毎年刊行される「現代用語の基礎知識」などの辞典類を見てもわかるように，比較的よく使われる言葉というものもどんどん新しくなっています。日本語としての用法や伝統的な語句については広辞苑が定評のあるところです。しかし，国語辞典というのもそれだけではありません。

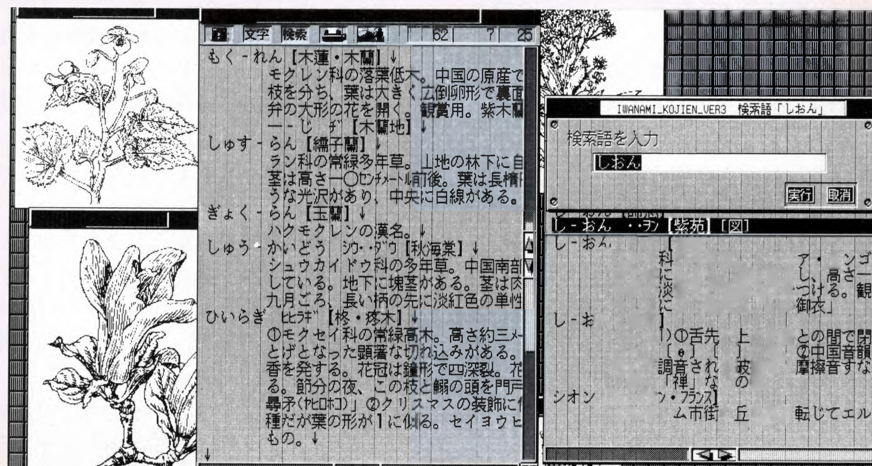
角川の類義語辞典や現代用語の基礎知識もCD-ROM化されていますので，同様な検索ソフトが現れればもっと多様なデータや最新の用語にでもデータアクセスできるようになるでしょう。

きたるべきマルチウィンドウ上のデータベースシステムにとってこのSX広辞苑はよい叩き台になるのではないかと思います。いずれはCD-ROMも当たり前の時代がくることは明らかです。広辞苑というのはそのようなとき欠かせないソフトのひとつだと思われますから。

また，発売バージョンでもまだ多少バグがあり，検索できない語や図版などが存在しました。安定性もイマイチです。計測技研では現在対応中ですので今後のサポートに期待しましょう。

**X68000用5″，3.5″2HD(デュアルメディア)**
12,800円(税別)
計測技研　☎0286(22)9811

図1　マルチウィンドウでの作業



# ついに発進！ サンダーワード1号

Nakano Shuichi 中野 修一

構想ン年，満開製作所が作り上げたサンダーワード１号。それはスマートで高速，偵察任務に向いたX68000用日本語ワードプロセッサだ。あらゆる装備を搭載できる２号もぜひ発売してほしいものだ。

THUNDERWORD

## BTRONに続け！

サンダーワード１号（以下雷語と略）という名前が初めて世に登場したのは1990年９月号の特集「日本語を処理するための序章」での祝一平氏の記事中でした。

そこで描かれているワープロと今回の雷語では若干の違いがあるものの，共に，

・microEmacs準拠の操作
・非オーバーラップマルチウィンドウ

といった，ほかの日本語ワープロソフトでは絶対に挙げられない特徴を堂々と掲げた製品です。

そもそも，その記事は，X68000での日本語環境がよくないということのみならず，ほぼありとあらゆるワープロソフトのあり方に対するアンチテーゼとしての意味を持っていました。

最近のワープロ事情を見てみると，

・アウトラインフォントによる美しい印字
・強力な日本語変換
・図形など，非文字情報の取り込み
・アウトラインプロセッサによる文書作成の支援

などの項目についての華々しい機能強化が見受けられます。これは，たいへん素晴らしいことです。

一方，雷語の機能を見てみると，

・出力は24ドットビットイメージのみ
・ASK68Kによる日本語変換
・扱えるのは文字情報のみ

アウトラインプロセッサにいたっては，「結局書き手の脳ミソのレベル以上の文章ができるわけないじゃないか」だそうで，流行のワープロと比べると雲泥の機能格差があることがわかります。いったいなんでいまさらこんな低機能なソフトを作るんだといぶかしがる人もいるでしょう。

ちょっとここで，視点を変えて考えてみましょう。

## 一太郎はなぜ嫌われるか？

バージョン５のリリースも間近に迫り，まもなく各パソコン誌の誌面は一太郎の話題で埋まってしまうはずです。日本語ワープロの代名詞のようになっている「一太郎」ですが，非常に嫌っている人が多いのも事実です。これはなぜでしょうか？

一太郎は，本格的なDTPはともかく，一太郎でできないことはないといっていいくらい充実した機能を持っています（ちょっといいすぎ）。

私もごくたまに一太郎に触ることがあります。単に文を打ち込むだけならどうということはないのですが，ちょっと立ち入ったことをやろうとすると，そのたびに驚かされます。コマンド体系がまったく理解の範囲を超えているからです。先ほどのワープロ強化項目のなかに，操作性というものがなかったことを思い出してください。

英文ワープロならば，機能や出力の美しさ以外に操作性という最重点項目が必ず入ってくることは間違いありません。むしろ，これ以外のものは付加機能の集まりにすぎないという見方もできます。

一太郎の場合，もっとも基本となる操作性の部分を改善しないまま複雑な機能を場当たり的に増設した感じに仕上がっている感じです。日本語の場合「変換」という困難ばかりが重視された結果，「エディット」ということに関してあまりにも無頓着なワープロが多いようです。

高度な機能を拡張することも大事かもしれませんが，土台となる部分が貧弱であればあるほど，飾りたてられたソフトは醜悪な様相を呈してきます。これが一太郎の嫌われる原因です。

同様の意味では，ほとんどすべての日本語ワープロソフトは失格であるといっていいでしょう。必要なのは多くの機能だけでなく，それが指先から自然に指示できるということです。

では，エディタ＋日本語FEPという組み合わせはどうでしょうか。テキストエディタというものは，本来，操作性を第一に考えて作成されています（良心的なものは）。

確かに現在では，多くの人が日本語FEP＋エディタという組み合わせで文章を書いています。ひと頃はひとつのトレンドと化していた気もします。

しかし，この組み合わせでは機能的な限界があることも事実です。ずっと以前，祝氏とエディタとワープロの違いについて話していたときエディタの機能的な限界として挙げられたのが「再変換」でした。FEPはあくまでFEPなのです。

## X68000の日本語環境

はっきりいって私はASK68K＋WP.Xの日本語環境にそんなに不満を持っていない人間です。なかには，あんなもので文章を書く人間の存在を信じられない人もいることと思います。私だってわけのわからない一太郎やMacintoshのつぶれた字で好んで文書を書く人間の存在は信じられませんが，世の中とはえてしてそういうものでしょう。

無論WP.Xにも欠点はあります。コントロールファンクションへの対応の仕方がいまいちなこと，行数表示の操作がうっとうしいこと，プリンタ設定がいまいちタコなこと，スクロールバーがイカレていること，禁則処理が不完全なこと，カーソルの移動が変なことなどの細かい点を挙げればキリがありません。最大の欠点は子プロセスが使えないことでしょう。そのくせ，メモリ全部を使って文書作成ができるかというと，さにあらず，100Kバイトを超えるような文書では，いくつかの章に分けて編集を行わなければならないのにも納得がいきません。しかし，それなりに長所もあり，つきあい

方を把握してしまえば，ほとんど不自由はありません。WP.Xでセーブ前の文書が暴走したときデバッガでメモリダンプをリダイレクトしてファイルを復活したりしたことも，いまとなっては懐かしい思い出です。

ASK68Kもなかなか「ユニーク」な日本語処理を展開してくれます。しかし，標準ではどうしようもなかった辞書も年月を重ねて使い込んでいくうちに，そこそこ使える辞書に成長します。いまでは，最初から「正しく変換するはずがない」と疑ってかかっていますから，他人が世間一般でASK68Kよりずっと賢いといわれる日本語FEPを使って作成した文書より，誤変換の混在率は少ないと思っています。

と，前置きしたところでX68000の日本語環境を振り返ってみましょう。

最初にWP.Xの対抗品として現れたソフトとしてEWという製品がありました。これの日本語FEPはE1というもので，VJEにほぼコンパチといわれていたものでした。

処理が遅い
とにかくよく暴走する
たいして変換効率がよくない

という3重苦によりほとんどユーザーはいなかったのではないかと思われます。数年前まで古村君が愛用していた以外に使っている人は見たことがありません。

続いて，シャープのHyperword，Multiwordなどですが，両者ともストレスを感じさせるほど処理が遅いという難点がありました。Hyperwordは一時期荻窪圭氏が愛用していたことでも有名ですが，マルチウィンドウにアイデアプロセッサといった近代風の特徴を持ち，正常進化していればかなり面白い存在になったはずです。

Multiwordは，これもイカレた操作体系の標本のようなソフトで，機能てんこ盛りのわりに使えない，使われていないという雷語の対極をいくものでした。

もちろん，

ED.X
microEmacs
SX-WINDOWのエディタ.X

などでも文書が書けますが，総合的にいってまだWP.Xのほうが使いやすい感じです(異論はあるでしょうが)。

さらにいえばワープロやエディタ以外のものでも文章を書くことはできるのですが，統合型表計算ソフトにせよ通信ソフトにせよ常駐型デスクアクセサリにせよ，テキスト編集機能は悪意を持って作られたのではないかと思えるくらいお粗末なものしか付属していませんでした。

## 雷語の特徴

すでに試用版を使ったことのある人も多いと思います。製品版はレスポンスや安定性ではひと味違うものの，基本的には同じものと考えてかまいません。

先ほども挙げたように雷語の最大の特徴はコマンド体系がエディタにあわせてあるという点です。それもX68000の標準スクリーンエディタED.Xコンパチなものと，パソコン上では最強と思われるmicroEmacsコンパチなものの2つが用意されています。

X68000ユーザーは大別してED派とmicroEmacs派に分類できます。ED.Xは一応使いものになるエディタですし，EDにアッパーコンパチなエディタとともに一大勢力を築いています。X68000版のmicroEmacsは凄まじいローカライズのおかげで強力さにさらに磨きがかかっており，第2の標準エディタと化した感があります。

たまにエディタ.XやWINDEXを使っているという人もいますが，FINALを使っているという人はまずいません。

ちなみに私はED派です。よって，以下の文章はEDモードを基準に記述していきます（本当はmicroEmacs派の人のほうが雷語のウケがいいんですけど）。

EDモードでは行単位だった範囲削除や張り付けが文字単位に拡張されていたり，ESC＋回数＋UNDOでキーボードマクロの連続実行ができたり，置換文字列中に改行を含むことができたり，ctrl-Kでもカットバッファに入ったり，という具合にいくつかの改良がなされています。画面まわりなどを含めて，microEmacsをEDバインディングで使用しているような感じですね。スクロールは標準のED.Xより心持ち速めです。

エディット環境自体はテキストエディタと同様なのですが，その他の部分をワープロとして見ると機能は決して高くありません。かろうじてアンダーラインはあるものの，罫線や拡大文字に対応しておらず，装飾類は皆無といっていいでしょう（個人的にはいらないが)。プリンタへの出力も24ドットビットイメージしか対応しておらず，いまのところ「文書を打ち込むだけ」といった性格の非常に強いワープロとなっています。機能長大化を続ける一般のワープロとはまったく異なったコンセプトです。

しかし，きわめてエディタに近い機能のものでも，再変換，完全にインラインでの変換作業といったワープロの環境を実現しています。変換ウィンドウが開くと文書の流れが読みにくくなりますし，目障りです。変換行が画面下に固定されている方式では注視点の移動が大きく，とても快適な日本語環境とはいえません。

雷語では視点は作成する文の流れに，指

### 祝！ サンダーワード発売

私が雷語で気に入っているのは，日本語入力を日本語変換FEP（以下FEPと略す）に頼っていないということである（変換部はともかく)。

私は雷語に出合う前はmicroEmacsとFIXER（またはASK68K)で原稿を書いていた(その前はWP.X)のだが，それのなにが頭にきたといって，編集コマンドを使いたいときにはASKをいったんキャンセルすることを強要されたことである。たとえばFEPをオンにして日本語の文章を入力している途中にキーボードマクロを使いたくなったとしよう。そこで手はするすると「CTRL＋X，(」を押すのであるが，最下行のコマンド表示エリアには「[CTRL-X]」の文字だけが空しく表示され，エディタのカーソルのそばに全角の「(」が未確定文字列となって表示される。ここで私は「きーっ」となってしまうのだ。

キーボードマクロを使おう，と心に決めて「CTRL＋X」を押した瞬間に，ユーザーの心理は編集コマンドのモードに入っているのに，FEPのほうは相変わらず律儀に文字コードを受け取って変換しようとしている。このズレから生じるイライラ。

念のためにいっておくが，これはmicroEmacsの責任でもFEPの責任でもない。しいていえば，FEPというカラクリそのものに限界があるということである。FEPは，キーボードの入力を横取りして日本語文字列に変換してエディタに渡す。FEPは，日本語入力とエディタの処理を切り離すことに成功したという点で，実に秀逸なアイデアではあるが，それゆえにエディタの動作状況などおかまいなしに動作してしまうのである。

そんなことはFEPの動作モードを常に意識しておけばすむ問題だ，とは思わない。たとえは悪いが，乗用車のクルーズコントロールはブレーキを踏めば自動解除するではないか。そうしないと事故を防げないからだ。

で，雷語であるが，これはASKのファンクションコールを使ってかな漢字変換の部分でASKの機能を利用している。キー入力の部分は雷語が受け持っている。

UNIXの世界ではちょっと名の知れた「NEmacs＋Wnn＋Egg」の組み合わせに使用感が似ている。NEmacs上の日本語入力モードを提供するEggは，かな漢字変換サーバであるWnnを利用する。これを最初に使ったときにはたいそう衝撃を受けたものだ。

そして雷語は，自動的に字詰めを行ってくれる点でNEmacsより優れている。また，鍛えた辞書を選べばASKはWnnよりはるかに賢い変換をしてくれる。結論として，雷語は現在私の使えるもっとも優れた日本語入力環境ということができよう。 (A.T.)

先はホームポジションに完全に固定して作業することができます。これは，本来，ごく当たり前の仕様なのですが，当然のことができていないワープロがいかに多いかを思い知らされます。

この当たり前の機能を実現するため，雷語ではASK68Kのワークエリアを直接参照しています。ゆえに，使用できる日本語変換FEPはASK68K.SYS ver.2.0のみに限定されています。FIXERがまだちゃんとFPコールに対応していない以上，これはしかたないことでしょう。

画面表示の際に行間を空ける設定ができるということも注目すべきです。文書は画面にめいっぱい詰めてしまうと非常に読みづらいものなのです。たとえば，このOh!Xの誌面では文字の縦方向と横方向の間隔比は3：2となっています。コンピュータの画面は文字が密着するように設計されていることが多いのですが，X68000ではビットマップ表示を採用しているため文書の流れを生かした表示方法が選べます。特にWP.Xに慣れてしまうとぎゅっと詰まった画面にはうんざりしてしまいます。

もちろん，このようにすると表示できる行数が多少減りますが，X68000ならこれでもPC-9801のワープロと同等の情報量を表示することができます。まず問題はないでしょう。まあこのあたりは個人的な好みが出てくるところですから，行間空きと詰めの両者を選択できる現在の方式は妥当な解決策だといえます。

その他の機能として挙げられる，キー割り当ての変更や普通のエディタとして使うエディタモードの搭載などもこのワープロの柔軟な性格を表しています。

## 信頼性

一般的にいって，日本語の文書を作成することはそんなに楽なことではない，と思われます。だからワープロというものは，絶対暴走しちゃいけない（当たり前か）ソフトのひとつといえるでしょう。

初期のWP.Xはよく飛びました。本当によく飛んだものでした。それもワープロとASK68Kのバージョンアップを重ねるごとに安定し，最近ではほとんど飛ばなくなってきました。だから比較的安心して使用していられるのですが，ソフトの信頼性を上げるには使い込むことが必要です。

こと日本語関係では「PC-9801のほうがMacintoshより優秀」という一見まるで根拠のなさそうな論調も，信頼性という点を加えると確かにうなずけるところがあります。

雷語は試作版を電脳倶楽部上で配布することでモニタリングを繰り返してきました。FRSなどを見てもわかるとおり，ユーザーの手で磨かれた開発ツールなどはメーカー製よりはるかに信頼性が高くなっています。多くの人を相手にチェック&デバッグをすることがバグをなくす近道です。

無論，雷語のマニュアルも雷語で書かれています。

レビュー用として，一般に公開された最終版をさらに煮詰め「もうほとんどバグはありません」と豪語していた最初のバージョンをもらい，バグレポートと改訂版の交換の日々が始まりました。開発者がほとんどやらないような操作ではいくつかバグが残っていましたが，それらがフィックスされるにつれ，普通に使って暴走させることはまずないと思われるくらいにはなってきました。

どうしても遅かった変換まわりのキー反応が改善されたことにより，高速に打ち込んだ場合に最後の母音を取りこぼす症状もなくなりました（たぶん）。

しかし，EDモードでいろいろいじめていた結果，結構信頼して使っていたED.Xのバグまで見つけてしまいました。あーあ。

## 雷語の課題

基本的な部分はかなりまとまっているのであとは付加機能の問題になります。

X68000の画面は横96文字（漢字48文字）分あります。しかし，普通の文書では48字詰めを使用することはほとんどないでしょう。この原稿は19字詰めで書いているわけですが，画面の右半分はまったく無駄になっています。雷語ではウィンドウの横分割をサポートしています。でも，日本語の文書の場合，縦分割のほうが有効なのは確かでしょう。

技術的な問題はほとんどないはずです。「テキスト → スクロール → ラスターコピー」というX68000での鉄壁の黄金律に対しては「テキストは4プレーンあるじゃないか」と布石を打っておきましょう。いずれにせよ，ver.1.1では縦分割をサポートすると断言していましたので，安心して待っていましょう。

変換候補文字列のエディットに制限が多いことも気になります。普通にASK68Kをコマンドラインから使用した場合に比べ，バックスペースによる文字削除しかできないのは少しさみしく思えます。まあ，この仕様に関しては私に責任があるような気もしないではないのですが……（昔のバージョンのバグでケチをつけたことがある）。

そのほか，章の概念がないので長い文書は作りにくいとか，やはり行数表示はリアルタイムでやってほしいとか，罫線くらいは必要だよなとか，縁起ものだからファイルセレクタくらいついていてもいいんじゃないかとか要望はいろいろありますが，いずれにせよ，基本的な日本語入力部分ができあがってしまえば，あとはごく付随的な問題にすぎません。今回のバージョンでは「日本語入力ツール」としての役割しか考慮されていないことは明らかです。その意味では素性のよいしっかりした地盤を作り上げたと評価していいでしょう。

今後雷語が本格的な「文書作成ツール」として成長することを期待しましょう。

## 最後に

この原稿は基本的に雷語で書かれたものですが，最終段階になってWP.Xに持ってきてしまいました。思いつくままにいろんなテーマで断片的に文章を書き，最後にまとめるというスタイルで原稿を仕上げているので，いまやWP.Xの章編集は私にとってなくてはならないものになっているのです。

雷語でも別のバッファを開けばマルチテキストが使えるのですが（むしろWP.Xより柔軟だとは思う），セーブ時には別ファイルになってしまうので気軽に章を増やしていくと再エディットのときに非常に面倒なことになってしまうのです。

雷語はまだ生まれたばかりのワープロです。このバージョンが機能的にも十分なものだとは思いません。逆にいえば，「まだ余計なものはなにもついていない」のです。そして，今後どのようなかたちの付加機能を加えていくかは，おそらくユーザーが決めていくことになるのでしょう。

とりあえず，ちゃんとバージョンアップしていくことはわかっていますから，使いにくいと思った部分はどんどん文句をつけていくことが重要です。強く主張すれば，作者の意図に反しない限り改善されていくでしょう。「いいワープロがほしい」と漠然と思っているだけでは状況は改善されないのです。

**THUNDER WORD**
**X68000用3.5/5"2HD版　20,000円(税別)**
**満開製作所　☎03(3554)9282**

Oh!X

# LIVE in '93

X68000・Z-MUSIC用
(SC-55対応)
美少女戦士
セーラームーンより

## ムーンライト伝説

Okamoto Masakazu
岡本 正和

X68000XVI・Z-MUSIC用

## チャコの海岸物語

Kameda Mineyuki 亀田 峰之

あけましておめでとうございます。さあ，LIVE in '93の幕開けです。今年最初の曲は，いまや人気絶頂のアニメ，あの「セーラームーン」のオープニング「ムーンライト伝説」と，X68000XVI用にカラオケの定番(?)「チャコの海岸物語」です。

### ああ，タキシード仮面さま♡

なんだかなぁ。今月もぶっとびでスタートしてしまいました。そう，美少女戦士セーラームーンです。新年早々おめでたいったらありゃしない。

曲はオープニングに流れている「ムーンライト伝説」です。テレビではDALIが歌っていますが，ほかのグループも歌っているとかいないとか。

演奏にはZ-MUSICシステムとZPP.XとSC-55が必要なのです。がっかりしないでね。

この作品では内蔵FM音源とSC-55の両方から音が出ています。ちゃんとミキシングしてくださいね。なんといってもSC-55を使わせたらセーラービーナスにも負けないといわれる(?)岡本君が作ったのですから。前回やその前の作品を入力した人ならわかるでしょう。

ZPP.Xは入力しましたか？　まだの人は早く入力したほうがいいですよ。1992年10月号に掲載されています。その効果は確かなものですし，これから投稿も増えていくと予想されます。ちなみに，今回のセーラームーンではZPPファイルで約22Kバイト，ZMSファイルに展開されると約33Kバイトとかなりオトク。

それから今回の小見出しはリクエストがあったのですが，ボツにしてしまいました。ゴメンナサイ。たとえば，「わたし，月野うさぎ！」とか「月の光に導かれ」，「悪霊退散！」だと。岡本君も完全に道を踏み外してしまったようですね。合掌。

ちなみに私はセーラームーンのBGMをほとんど口ずさめます。すべてはコメントをほとんど書かなかった蓮沼君(1992年10月号)のおかげです。岡本君はとっても読みごたえのあるドキュメントをつけてくれました。どうもありがとう。

### 要15MHz!?

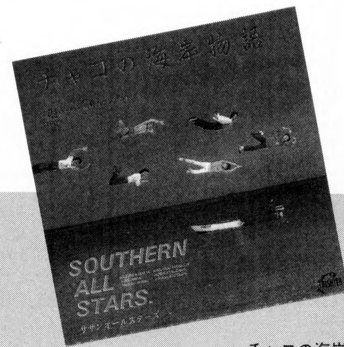
さて，X68000XVI+Z-MUSICシステム+PCM8.X用に「チャコの海岸物語」をお届けしましょう。市販のソフトでもXVI専用なんてないのに，なぜXVI専用なのかと申しますと，10MHzでは演奏が正常に行われないからなのです。

説明しましょう。PCM8.Xを使用したときのX68000は非常に重くなることを知っていますよね。「重くなる」といっても，別に重量が2tになるとかじゃないですよ。動作が遅くなるって意味です。このときのMPU68000の実動作速度は40%とかに低下しているのです。つまり，10MHzのマシンでは4MHz程度に落ちているわけです。残りのパワーは主にPCM8.Xがサンプリングデータを加工するのに使われているのです。そこでPCM8.Xに多量のデータを渡すと，サンプリングデータの加工だけでMPU68000が手いっぱいになってしまい，演奏ができなくなったり，サンプリングデータの加工すら間に合わなくなったりします。

1992年の7月号にも「正統派で邪道なテクニック(?)」として亀田君の「Bye Bye My Love」が掲載されていますが，あの作品では曲の途中で20%程度までパワーダウンし，最後のフェードアウトでは10%程度までパワーダウンしていたのです。フェードアウトがヘンだと思った10MHzユーザーもいたことでしょう。ところが，この「チャコの海岸物語」では5%は当たり前，最低では0%までパワーダウンしてしまいます。さすがに3%を割ったときは演奏になっていません。なんといっても数百KHzの世界ですから。



美少女戦士セーラームーン



チャコの海岸物語

説明が長くなってしまいましたが，以上の話は10MHzのマシンに限ったこと。XVIなら問題なく演奏されます。亀田君はX68000に改造を加えているようで，15MHzならばちゃんと演奏できるそうです。今月号の46ページで紹介しているPOLYPHONを使用するという手もありますね。

さて，曲のデキはといえば「XVI専用でも掲載されるくらい」といえばおわかりいただけるでしょうか。レーザージュークを

聴いているみたいです。

演奏に関する注意点として，サンプリングデータが500Kバイトにもなりますので，

ZMUSIC -P500

としてバッファを確保する必要があります。また，ZPCNVでサンプリングファイルを作るときも空きメモリを十分にとってください。Z-MUSICシステムはできるかぎり最新のものを使ったほうがいいでしょう。

実は，同封の「モーニングムーン要15MHz」も素晴らしかったのですが，どうせならみんなで聴ける10MHz対応バージョンを掲載しようとしたのです。ところが，ファイルがCRCエラーを起こしていて読み取り不可能だったのです。亀田君，ロークロックバージョンを「ちゃんと2度セーブして」もう一度送り直してください。それから，「チャゲ＆飛鳥のモーニングムーンが聴きたいぞ！」というリクエストも受け付けます。10MHzバージョンが届きしだい，検討したいと思います。

## ごめんなさい

またもやポカをやってしまいました。先月号で掲載されていたWHAM！の「LAST CHRISTMAS」の音色データが抜けていました。本当にご迷惑をおかけしました。今月号で掲載したダンプリストを入力して，LHAで展開してください。　(S.K.)

### リスト1　ムーンライト伝説

日本音楽著作権協会(出)許諾第9272282-201号

```
.comment 美少女戦士セーラームーン ~ムーンライト伝説~ ..by MAz
.comment midi-module  SC-55  プログラム  岡本正和


(i)(b1)(d1)


(m01,2000)(aMidi10,01)
(m02,2000)(aMidi10,02)
(m03,1000)(aMidi10,03)
(m04,2000)(aMidi10,04)

(m06,2000)(aFm1   ,06)
(m07,2000)(aFm2   ,07)
(m08,2000)(aFm3   ,08)
(m09,2000)(aFm4   ,09)

(m20,2000)(aFm5   ,20)
(m21,2000)(aFm6   ,21)
(m22,2000)(aFm7   ,22)

(m30,3000)(aMidi02,30)
(m31,3000)(aMidi11,31)
(m32,3000)(aMidi03,32)

(m40,4000)(aMidi01,40)
(m41,4000)(aFm8   ,41)

(m50,2000)(aMidi07,50)

(m60,3000)(aMidi04,60)
(m61,3000)(aMidi05,61)
(m62,3000)(aMidi06,62)

(m70,2000)(aMidi08,70)
(m71,2000)(aMidi09,71)


.SC55_PRINT " THE LEGEND of MOON LIGHT "


(t1) @I$41,$10,$42
(t1) X$40,$00,$7f,0
(t1) X$40,$01,$30,3
(t1) X$40,$01,$34,85
(t1) X$40,$01,$38,1
(t1) X$40,$01,$10,   0, 1,1,1, 2,2,2,2, 2,2,1,0, 0,0,0,0


(v1,0

          6, 15,  3,  0,255,127,  0,  6,  0,  2,  0

         31,  0,  0,  6, 15,  0,  0,  4,  0,  0,  0
         31, 15,  0,  6, 15,  5,  0,  2,  0,  0,  0
         31, 15,  0,  6, 15,  0,  0,  2,  3,  0,  0
         31, 15,  0,  6, 15,  0,  0,  8,  7,  0,  0)

(v2,0

         45, 15,  3,  0,255,127,  0,  7,  0,  3,  0

         31, 20, 13, 15,  7, 10,  0, 15,  0,  0,  0
         31, 13, 13, 15,  8,  0,  0,  0,  0,  0,  0
         31, 13, 13, 15,  8,  0,  0,  0,  0,  0,  0
         31, 12, 13, 15,  8,  0,  0,  0,  0,  0,  0)

(v3,0

         40, 15,  3,  0,255,127,  0,  0,  0,  3,  0

         31,  4,  1, 15,  1, 10,  0,  2,  0,  0,  0
         29,  1,  1, 15,  1,  8,  0,  5,  1,  0,  0
         31,  4,  1, 15,  1, 23,  0,  4,  1,  0,  0
         31, 12,  2, 15,  1,  0,  0,  1,  2,  0,  0)

(v4,0

         61, 15,  3,  0,255,127,  0,  7,  0,  3,  0

         31, 26, 13, 10,  8, 30,  0,  0,  0,  0,  0
         31, 16, 13, 10,  8,  0,  0,  3,  0,  0,  0
         31, 16, 13, 10,  8,  0,  0,  3,  0,  0,  0
         31, 16, 13, 10,  8,  0,  0,  3,  0,  0,  0)

(v5,0

         61, 15,  3,  0,255,127,  0,  7,  0,  3,  0

         31, 26, 13, 10,  8, 30,  0,  0,  0,  0,  0
         31, 11, 13, 10,  8,  0,  0,  3,  0,  0,  0
         31, 12, 13, 10,  8,  0,  0,  3,  0,  0,  0
         31, 13, 13, 10,  8,  0,  0,  3,  0,  0,  0)

(v6,0

         45, 15,  3,  0,255,127,  0,  7,  0,  3,  0

         31, 20,  5,  6,  7, 10,  0, 15,  0,  0,  0
         31, 13,  5,  6,  8,  0,  0,  2,  0,  0,  0
         31, 13,  5,  6,  8,  0,  0,  1,  0,  0,  0
         31, 13,  5,  6,  8,  0,  0,  1,  0,  0,  0)

(v7,0

         40, 15,  3,  0,255,127,  0,  0,  0,  3,  0

         31,  4,  1,  0,  1, 10,  0,  3,  4,  0,  0
         18,  1,  1,  8,  1, 28,  0, 15,  4,  0,  0
         31,  4,  1,  0,  1, 23,  0,  7,  7,  0,  0
         31, 12,  2,  8,  1,  0,  0,  1,  7,  0,  0)

(v8,0

         50, 15,  3,  0,255,127,  0,  0,  0,  3,  0

         30,  4,  2,  6,  2, 37,  1,  0,  3,  0,  0
         31,  8,  0,  8, 12, 46,  1,  5,  5,  0,  0
         16,  5,  2,  6,  2, 21,  0,  0,  3,  0,  0
         11,  5,  0,  6,  0,  0,  0,  0,  5,  0,  0)

(v9,0

         27, 15,  3,  0,255,127,  0,  5,  0,  3,  0

         31, 10, 20, 15, 11, 20,  1,  0,  1,  0,  0
         31,  3, 20, 15,  4, 40,  0,  3,  5,  0,  0
         31,  3, 20, 15,  4,  0,  2,  0,  3,  0,  0
         31,  3, 20, 15,  5,  0,  0,  1,  0,  0,  0)

(v10,0

         59, 15,  3,  0,255,127,  0,  0,  0,  3,  0

         31,  2,  1,  3,  1, 30,  0,  7,  0,  0,  0
         31,  2,  1,  3,  1, 40,  0,  4,  7,  0,  0
         31,  3,  2,  3,  1, 40,  1,  2,  3,  0,  0
         31,  4,  3,  7,  1,  0,  1,  1,  0,  0,  0)


(t1)    t141

(t30)   r*6
(t30)   y126,1
(t30)   @11y91,50y93,30q6v15@g12u11018r4
(t30)   r4r1
(t30)   r2^g<ce-
(t30)   gr
(t30)   gfrfe-r
(t30)   d
(t30)   r
(t30)   fr^>b<df
(t30)   a-ra-b-ra-gfe-
(t30)   r
(t30)   g
(t30)   r^crg
(t30)   <c
(t30)   r
(t30)   >g<c>b
(t30)   r
(t30)   gb
(t30)   b-r
(t30)   a-r^a-gfgrfre-drc
(t30)   r2^@72y91,50y93,60v14
(t30)   q7>g<c&@b2048c
(t30)   q7@b0g^gf^fe-e-q5d^f^rq7>gb<df^fe-^e-d^c^q6e-^rq
7>g<c&@b2048c
(t30)   q7@b0g^gf^fe-e-q6d^f^rq7>gb<df^gfe-d^d&@b682d4.r
```

```
4q7@b0>g<c&@b2048c
   168: (t30)   q7@b0g^gf^fe-^q8d^q6f^rq7>gb<df^ffe-d^c^^q6e-^rq
7>g&q5@b3413gq7@b0<e-
   169: (t30)   g^gf^fggq8b-&@b0,-1365b-@b0a-r^q7a-&@b-682a-@b0f
g^f&_12f~e-d^q8c^^r2c^
   170: (t30)   a-2.b-b-&q6@b1365b-^q8@b0<c^q7>b-a-^^g^^fe-f^q8g
^2r^c^
   171: (t30)   a-2.b-b-&q6@b1365b-^rq7@b0<c{c>b-a-}2g1&¥50g2¥0~
0rq7>g<ce-
   172: (t30)   q7g^gf^fe-e-q5d^f^rq7>gb<df^gfe-q5d^q6c^^q5e-^rq
7>g.
   173: (t30)   <c&@b2048c@b0g^gf^fggq8b-&@b-1365b-@b0a-r^q7a-gf
q5g^f^q8~16e-q5d^q8c^^
   174: (t30)   r2.
   175: (t30)   @11y91,50y93,30q814_a-fge-fde-cr1
   176: (t30)   @72y91,50y93,60
   177: (t30)   18r2^q7>g<ce-
   178: (t30)   q7g^gf^fe-^q6d^f^rq7>gb<df^ffe-e-q5d^q8c^q6e-^rq
7>g<ce-
   179: (t30)   q7g^q8gq7f^fg^b-&@b-1365b-@b0a-^ra-gfg^fq8fq5e-q
7d^c^^r2c^
   180: (t30)   q8a-2.q7b-b-&@b1365b-^@b0<c^>b-a-^^g^^q6fe-q7f^q
8g^2r^q7c^
   181: (t30)   q8a-2.q7b-b-&q8@b1365b-^rq7@b0<c{c>b-a-}2g1&¥50g
2¥0~0rq7>g<ce-
   182: (t30)   q7g^gf^fe-e-q5d^f^rq7>gb<df^gfe-q5d^q6c^^q5e-^rq
7>g
   183: (t30)   <c&@b2048c@b0g^gf^fggq8b-&@b-1365b-@b0a-r^q7a-gf
q5g^f^q8~16e-q5d^q8c^^
   184: (t30)   @11y91,50y93,30v15
   185: (t30)   q6r^^>g<ce-
   186: (t30)   gr
   187: (t30)   gfrfe-r
   188: (t30)   d
   189: (t30)   r
   190: (t30)   fr^>b<df
   191: (t30)   a-ra-b-ra-gfe-
   192: (t30)   r
   193: (t30)   g
   194: (t30)   r^ce-g
   195: (t30)   <c
   196: (t30)   r
   197: (t30)   >g<c>b
   198: (t30)   r
   199: (t30)   gb
   200: (t30)   b-r
   201: (t30)   a-r^a-gfgrfre-drc
   202: (t30)   r2^@72y91,50y93,60v14
   203: (t30)   rq8c^
   204: (t30)   q8a-2.q7b-b-&@b1365b-^@b0<c^>b-a-^^g^^q6fe-q7f^q
8g^2r^q7c^
   205: (t30)   q8a-2.q7b-b-&q8@b1365b-^rq7@b0<c{c>b-a-}2g1&¥50g
2¥0~0rq7>g<ce-
   206: (t30)   q7g^gf^fe-e-q5d^f^rq7>gb<df^gfe-q5d^q6c^^q5e-^rq
7>g
   207: (t30)   <c&@b2048c@b0g^gf^fggq8b-&@b-1365b-@b0a-r^q7a-gf
q5g^f^q8~16e-q5d^q8c^^
   208: (t30)   r2q7_gf
   209: (t30)   q5g^f^q8e-q5d^q8c^^r4.q7a-gfq5g^q4a-^q8aq4b^<c^r
*22
   210: (t30)   @62
   211: (t30)   v15q8>>>c1^^q5>c^
   212: (t31)   r*6
   213: (t31)   y126,1
   214: (t31)   @73y91,70y93,30q6v8u110@p818@g12r4
   215: (t31)   r4r1
   216: (t31)   r2^<g<ce-
   217: (t31)   gr
   218: (t31)   gfrfe-r
   219: (t31)   d
   220: (t31)   {de-e}
   221: (t31)   fr^>b<df
   222: (t31)   a-ra-b-ra-gfe-
   223: (t31)   {e-ef}
   224: (t31)   g
   225: (t31)   r^ce-g
   226: (t31)   <c
   227: (t31)   >e-
   228: (t31)   g<c>b
   229: (t31)   e-
   230: (t31)   gb
   231: (t31)   b-r
   232: (t31)   a-r^a-gfgrfre-drc
   233: (t31)   r2^@18y91,50y93,60p3@k3v12
   234: (t31)   q7>g<c&@b2080c
   235: (t31)   q7@b32g^gf^fe-e-q5d^f^rq7>gb<df^fe-^e-d^c^q6e-^r
q7>g<c&@b2080c
   236: (t31)   q7@b32g^gf^fe-e-q6d^f^rq7>gb<df^gfe-d^d&@b714d4.
r4q7@b32>g<c&@b2080c
   237: (t31)   q7@b32g^gf^fe-^q8d^q6f^rq7>gb<df^ffe-d^c^^q6e-^r
q7>g&q5@b3445gq7@b32<e-
   238: (t31)   g^gf^fggq8b-&@b32,-1333b-@b32a-r^q7a-&@b-650a-@b
32fg^f&_12f~e-d^q8c^^r2c^
   239: (t31)   a-2.b-b-&q6@b1397b-^q8@b32<c^q7>b-a-^^g^^fe-f^q8
g^2r^c^
   240: (t31)   a-2.b-b-&q6@b1397b-^rq7@b32<c{c>b-a-}2g1&¥50g2¥0
~0rq7>g<ce-
   241: (t31)   q7g^gf^fe-e-q5d^f^rq7>gb<df^gfe-q5d^q6c^^q5e-^rq
7>g
   242: (t31)   <c&@b2080c@b32g^gf^fggq8b-&@b-1333b-@b32a-r^q7a-
gfq5g^f^q8~16e-q5d^q8c^^
   243: (t31)   r2.
   244: (t31)   @11y91,50y93,30q814_a-fge-fde-cr1
   245: (t31)   @18y91,50y93,60
   246: (t31)   18r2^q7>g<ce-
   247: (t31)   q7g^gf^fe-^q6d^f^rq7>gb<df^ffe-e-q5d^q8c^q6e-^rq
7>g<ce-
   248: (t31)   q7g^q8gq7f^fg^b-&@b-1333b-@b32a-^ra-gfg^fq8fq5e-
q7d^c^^r2c^
   249: (t31)   q8a-2.q7b-b-&@b1397b-^@b32<c^>b-a-^^g^^q6fe-q7f^
q8g^2r^q7c^
   250: (t31)   q8a-2.q7b-b-&q8@b1397b-^rq7@b32<c{c>b-a-}2g1&¥50
g2¥0~0rq7>g<ce-
   251: (t31)   q7g^gf^fe-e-q5d^f^rq7>gb<df^gfe-q5d^q6c^^q5e-^rq
7>g
   252: (t31)   <c&@b2080c@b32g^gf^fggq8b-&@b-1333b-@b32a-r^q7a-
gfq5g^f^q8~16e-q5d^q8c^^
   253: (t31)   @73y91,70y93,30v8@p8
   254: (t31)   q6r^^>g<ce-
   255: (t31)   gr
   256: (t31)   gfrfe-r
   257: (t31)   d
   258: (t31)   {de-e}
   259: (t31)   fr^>b<df
   260: (t31)   a-ra-b-ra-gfe-
   261: (t31)   {e-ef}
   262: (t31)   g
   263: (t31)   r^ce-g
   264: (t31)   <c
   265: (t31)   >e-
   266: (t31)   g<c>b
   267: (t31)   e-
   268: (t31)   gb
   269: (t31)   b-r
   270: (t31)   a-r^a-gfgrfre-drc
   271: (t31)   r2^@18y91,50y93,60p3@k3v12
   272: (t31)   rq8c^
   273: (t31)   q8a-2.q7b-b-&@b1397b-^@b32<c^>b-a-^^g^^q6fe-q7f^
q8g^2r^q7c^
   274: (t31)   q8a-2.q7b-b-&q8@b1397b-^rq7@b32<c{c>b-a-}2g1&¥50
g2¥0~0rq7>g<ce-
   275: (t31)   q7g^gf^fe-e-q5d^f^rq7>gb<df^gfe-q5d^q6c^^q5e-^rq
7>g
   276: (t31)   <c&@b2080c@b32g^gf^fggq8b-&@b-1333b-@b32a-r^q7a-
gfq5g^f^q8~16e-q5d^q8c^^
   277: (t31)   r2q7_gf
   278: (t31)   q5g^f^q8e-q5d^q8c^^r4.q7a-gfq5g^q4a-^q8aq4b^<c^r
*22
   279: (t31)   @62
   280: (t31)   @k4
   281: (t31)   v15q8>>>c1^^q5>c^
   282: (t32)   r*6
   283: (t32)   y126,1
   284: (t32)   @11y91,50y93,30q6v12u110@p40@g1218r4
   285: (t32)   r*432
   286: (t32)   <e-re-drdcr>br<dr^>fb<d
   287: (t32)   frfgrfe-dcre-r^2r1r1
   288: (t32)   r1r1
   289: (t32)   @p120@73v11r1r1r1116r4.q8<e-fde-cd>b-<c>a-b-
   290: (t32)   b<cd-de-8r8^2r1r1@72y91,50y93,60@p7218r2^q7>>e-g
<c
   291: (t32)   q7e-^e-d^dc^q8>b^q6<d^rq7>dgb
   292: (t32)   <d^ddc>b^g^^q6<c^r2r1r1r1r2.q8>g^
   293: (t32)   <f2.gg&q6a-^q8a-^q7gf^^e-^^dcd^q8e-^2r^>g^
   294: (t32)   <f2.gg&q6a-^rq7a-{a-gf}2d1&¥50d2¥0~0r2
   295: (t32)   |:5r1:|r2^q7fe-dq5e-^d^q8_14gq5f^q8e-^^
   296: (t32)   r2.
   297: (t32)   p3@12v1112>a-gfe-18@72y91,50y93,60r1v11@p72r2^q7
e-g<c
   298: (t32)   q7e-^e-d^dc^q6>b^<d^rq7>dgb
   299: (t32)   <d^ddccq5>b^q8g^q6<c^r2r1r1r1r2.q8>g^
   300: (t32)   q8<f2.q7gg&a-^a-^gf^^e-^^q6dcq7d^q8e-^2r^q7>g^
   301: (t32)   q8<f2.q7gg&q8a-^rq7a-{a-gf}2d1&¥50d2¥0~0r2
   302: (t32)   |:5r1:|r2^q7fe-dq5e-^d^q8_14gq5f^q8e-^^
   303: (t32)   r2.
   304: (t32)   @11y91,50y93,30q6v12@p40
   305: (t32)   e-re-drdcr>br<dr^>fb<d
   306: (t32)   frfgrfe-dcre-r^2r1r1
   307: (t32)   v14g^a-^ab^<c^2.@72y91,50y93,60@p72v12q8>>g^
   308: (t32)   q8<f2.q7gg&a-^a-^gf^^e-^^q6dcq7d^q8e-^2r^q7>g^
   309: (t32)   q8<f2.q7gg&q8a-^rq7a-{a-gf}2d1&¥50d2¥0~0r2
   310: (t32)   |:5r1:|r2^q7fe-dq5e-^d^q8_14gq5f^q8e-^^
   311: (t32)   r2q7~e-d
   312: (t32)   q5e-^d^q8gq5f^q8e-^^r4.q7fe-dq5e-^q4f^q8e-q4d^c^
   313: (t60)   r*4
   314: (t60)   v14u110@63y93,80y91,6018r4
   315: (t60)   r4
   316: (t60)   18>g-q6gr2q8<e-q4dq8>bq4<dq8a-q4gr2
   317: (t60)   @v82
   318: (t60)   y126,1
   319: (t60)   |:
   320: (t60)   q418<c>g<c>g16&<cc16&>g
   321: (t60)   |<c>g|:
   322: (t60)   bgbg16&bb16&gbg
   323: (t60)   :|:|q8{<c>ge-c}4
   324: (t60)   q4ge-cc16&gg16&e-c>b-<fffr^v14fe-d
   325: (t60)   e-rdrc>br
   326: (t60)   <cr^q8>b2.
   327: (t60)   v13
   328: (t60)   |:7r1:|rq8g-q4gq8bq4<c_5q8bq4<cq8d
   329: (t60)   v10
   330: (t60)   q418c>gcc16e-e-16dc>g<bf>bb16<ff16d>bg
   331: (t60)   <bf>bb16<bb16f>d>b<<<c>gcc15gg16e-c>g
   332: (t60)   <<c>cg<c>bq8{ce-fe-}4q4ca-cfc16ff16c>a-f
   333: (t60)   <gra-rabr<c
   334: (t60)   y127,0
   335: (t60)   @49y93,00@p4r*140q8{cg<c>g}4&g*2
   336: (t60)   <c*190&c1&¥30c2^¥0~0>f^g^1<c1^r*172v14@63y93,801
8p3
   337: (t60)   r>>>g-q6grq8<e-q6drq8>b-bq4<dq8a-q4gr2
   338: (t60)   |:3r1:|@49y93,00@p4r*117q8>g<ce-
   339: (t60)   g^gf^fggb-a-a-r^q7a-gfq5g^f^q8e-q5d^q8
   340: (t60)   c*219
   341: (t60)   v14@63y93,8018p3r1r1
   342: (t60)   18>g-q6gr2q8<e-q4dq8>bq4<dq8a-q4gr2
   343: (t60)   y126,1
   344: (t60)   v10
   345: (t60)   q418<c>gcc16e-e-16dc>g<bf>bb16<ff16d>bg
   346: (t60)   <bf>bb16<bb16f>d>b<<<c>gcc16gg16e-c>g
   347: (t60)   <<c>cg<c>bq8{ce-fe-}4q4ca-cfc16ff16c>a-f
```

```
(t60)   <gra-rabr<c
(t60)   y127,0
(t60)   @49y93,00@p4r*140q8{e-fgb-}4
(t60)   a-2.b-b-<c1>g2^f^g^1a-2.b-b-<c^r*148v14@63y93,80
18p3
(t60)   r>>>g-q6grq8<e-q6drq8>b-bq4<dq8a-q4gr2
(t60)   |:3r1:|@49y93,00@p4r*117q8>g<ce-
(t60)   g^gf^fggb-a-a-r^q7a-gfq5g^f^q8e-q5d^q8
(t60)   c*219
(t60)   @63y93,80@v82p3
(t60)   y126,1
(t60)   q418<c>g<c>g16&<cc16&>ge-cbgbg16&bb16&gd>b
(t60)   <bfbf16&bb16&fd>b<<c>g<c>g16&<c>g16&e-c>g
(t60)   <ge-cc16&gg16&e-c>b-<fff
(t60)   @v82
(t60)   r^v14fe-de-rdrc>brq8
(t60)   <c*211
(t60)   v10@49y93,00p3a-1&¥35a-1¥0~0g1&¥35g1¥0~0a-1a-4r*
149
(t60)   @63y93,80v14
(t60)   y127,0
(t60)   r>g-q6grq8<e-q6drq8>b-bq4<dq8a-q4gr2
(t60)   |:3r1:|@49y93,00@p4r*117q8>g<ce-
(t60)   g^gf^fggb-a-a-r^q7a-gfq5g^f^q8e-q5d^q8
(t60)   c*219
(t60)   q5g^f^q8aq5b^q8<c*219@63y93,80q3v14>>>>g^r2.
(t60)   @49y93,00@p4v15r*43q8<<<<d1^^q5>c^
(t61)   r*4
(t61)   v12u110@30@g12y91,60y93,6018r4
(t61)   @b0,-4778>>a&@b-4778,-7509a
(t61)   v14u110@63y93,80k0
(t61)   18@b0<<g-q6gr2q8<e-q4dq8>bq4<dq8a-q4gr2
(t61)   v10
(t61)   y126,1
(t61)   q4|:
(t61)   >e-ce-c16&e-e-16&c
(t61)   |e-c
(t61)   q418d>b<d>b16&<dd16&>b<d>b<fdfd16&ff16&dfd
(t61)   :|q8{e-c>ge-}4q4c>ge-e-16&<cc16&>gc>e-
(t61)   <dddr^2v13gga-a-abb
(t61)   <cr^q8g2.
(t61)   v11q8|:7r1:|r>g-gb<c_5b<cd
(t61)   v14q5<e-r2..
(t61)   y127,0
(t61)   @49y93,00v16|:6r1:|r*91q8{>cde-ce-fge-}2
(t61)   v12f1&f1e-2^d^e-^1
(t61)   f1e-r*173
(t61)   v14@63y93,80
(t61)   rg-q6grq8<e-q6drq8>b-bq4<dq8a-q4gr2
(t61)   |:3r1:|@49y93,00@p90r*189q8>e-^e-d^de-e-gffr^q7f
e-d
(t61)   q5e-^d^q8gq5f^q8
(t61)   e-*219p3
(t61)   @63y93,80r1r1
(t61)   18g-q6gr2q8<e-q4dq8>bq4<dq8a-q4gr2
(t61)   r1
(t61)   y127,0
(t61)   @49y93,00v16|:6r1:|r*91q8{>cde-cce-fg}2
(t61)   v12f1&f1e-2^d^e-^1
(t61)   f2.gga-^r*149
(t61)   v14@63y93,80
(t61)   rg-q6grq8<e-q6drq8>b-bq4<dq8a-q4gr2
(t61)   |:3r1:|@49y93,00@p90r*189q8>e-^e-d^de-e-gffr^q7f
e-d
(t61)   q5e-^d^q8gq5f^q8
(t61)   e-*219p3
(t61)   v10@63y93,80
(t61)   y126,1
(t61)   q418e-ce-c16&e-e-16&c>ge-d>b<d>b16&<dd16&>bgd
(t61)   <fdfd16&ff16&d>bf<e-ce-c16&e-c16&>ge-c
(t61)   <c>ge-e-16&<cc16&>gc>e-<ddd
(t61)   @63y93,80y91,60
(t61)   r^2v13gga-a-abbq8<c*211
(t61)   v9@49y93,00<f1&¥35f1¥0~0e-1&¥35e-1¥0~0f1e-4r*149
(t61)   v14@63y93,80
(t61)   y127,0
(t61)   rg-q6grq8<e-q6drq8>b-bq4<dq8a-q4gr2
(t61)   |:3r1:|@49y93,00@p90r*189q8>e-^e-d^de-e-gffr^q7f
e-d
(t61)   q5e-^d^q8gq5f^q8
(t61)   e-^1
(t61)   q5e-^d^q8gq5f^q8e-*219@63y93,80q3v14>>g^r2.
(t61)   @49y93,00@p90v14r*43q8<<a1^^q5c4
(t62)   r*4
(t62)   @g12v13u110@49y93,50y91,6018r4
(t62)   r4
(t62)   18>>g-q6gr2q8<e-q4dq8>bq4<dq8a-q4gr2
(t62)   @63y93,80v12q8r1b{b<cd-}d4r2r1c{cd-d}e-4r*19
(t62)   q8@49y93,00@p16v14{<cd}{e-ce-f}4g^e-cgfe-c
(t62)   g^fr^
(t62)   y126,1
(t62)   v13q7>a-gfgrfre-drcr^q8>g2.r*5
(t62)   @49y93,50v12p3
(t62)   |:7r1:|rq8>g-q4gq8bq4<c_5q8bq4<cq8d
(t62)   v15q5e-r2..
(t62)   q8
(t62)   y127,0
(t62)   v11@49y93,00|:7r1:|
(t62)   a-*187&a-1g2^f^g^v13>g<ce-v14{<cde-c}4v12{g<cdc}
4
(t62)   v11>>a-2^v14{a-f}8{c>a-fd}4v13cr*173@49y93,50
(t62)   r>g-q6grq8<e-q6drq8>b-bq4<dq8a-q4gr2
(t62)   y126,1
(t62)   |:5r1:|p3@63y93,80r1q5g^a-^q8aq5b^<c^1
(t62)   r1@49y93,50
(t62)   y127,0
(t62)   r1
(t62)   18>>g-q6gr2q8<e-q4dq8>bq4<dq8a-q4gr2
(t62)   y126,1
(t62)   r1
(t62)   q8
(t62)   y127,0
(t62)   v11@49y93,00|:7r1:|
(t62)   <a-*187&a-1g2^f^g^v13>g<ce-v14{<cde-c}4v12{g<cdc
}4
(t62)   v11>>a-2^v14{a-f}8{c>a-fd}4v13cr*173@49y93,50
(t62)   r>g-q6grq8<e-q6drq8>b-bq4<dq8a-q4gr2
(t62)   y126,1
(t62)   |:5r1:|p3@63y93,80r1q5g^a-^q8aq5b^<c^1
(t62)   @63y93,80v13q8q7e-re-drd&cr
(t62)   y127,0
(t62)   _8>b{b<cd-}d4r~
(t62)   y126,1
(t62)   q8>fb<dq7frfgrq8fe-d
(t62)   y127,0
(t62)   _c{cd-d}e-4r*19q8@49y93,00@p16v14{<cd}{e-ce-f}4
(t62)   g^e-cgfe-cg^fr^
(t62)   y126,1
(t62)   v13q7>a-gfgrfre-drq8e-^1
(t62)   v9@49y93,00c1&¥35c1¥0~0c1&¥35c1¥0~0c1c4r*149
(t62)   v13@49y93,50
(t62)   y127,0
(t62)   r>>g-q6grq8<e-q6drq8>b-bq4<dq8a-q4gr2
(t62)   y126,1
(t62)   |:5r1:|p3@63y93,80r1q5g^a-^q8aq5b^<c^1
(t62)   q5e-^d^q8cq5>b^q8<c*211@49y93,00v15q5>g^r4.q8abq
5<c^
(t62)   rq8e-1^^q5>c^
(t50)   r*6
(t50)   v14u110@31y93,60y91,60@g1218r4
(t50)   @b0,-4778>a&@b-4778,-7509a
(t50)   v12
(t50)   18@b0>g-q6gr2q8<e-d>b<da-g
(t50)   r2
(t50)   q8@63y93,80v12r*190g{a-ab-}b4r2r1e-{efg-}g4r*91q
8@49y93,00v13
(t50)   e-4c.gg16fe-gb-4a-r^2q4gga-a-abb
(t50)   <cr^q8>g2.r*7
(t50)   v13@p8@31y93,60
(t50)   |:7r1:|rq8g-q4gq8bq4<c_5q8bq4<cq8d
(t50)   v9@30
(t50)   q418>>c>gcc16e-e-16dc>g<bf>bb16<ff16d>bg
(t50)   <bf>bb16<bb16f>d>b<<<c>gcc16gg16e-c>g
(t50)   <<c>cg<c>bq8{ce-fe-}4q4ca-cfc16ff16c>a-f
(t50)   q8v11<gra-rabr<c^r2..
(t50)   v12@31y93,60
(t50)   y127,0
(t50)   |:6r1:|
(t50)   r>g-q6grq8<e-q6drq8>b-bq6<dq8a-q6gr2
(t50)   y126,1
(t50)   |:5r1:|@63y93,80v12r1q5>g^a-^q8aq5b^<c^1
(t50)   @30v7@q3@p812fe-dc
(t50)   y127,0
(t50)   v12@31y93,60
(t50)   18>g-q6gr2q8<e-d>b<da-g
(t50)   r2
(t50)   y126,1
(t50)   v9@30
(t50)   q418c>gcc16e-e-16dc>g<bf>bb16<ff16d>bg
(t50)   <bf>bb16<bb16f>d>b<<<c>gcc16gg16e-c>g
(t50)   <<c>cg<c>bq8{ce-fe-}4q4ca-cfc16ff16c>a-f
(t50)   q8v11<gra-rabr<c^r2..
(t50)   v12@31y93,60
(t50)   y127,0
(t50)   |:6r1:|
(t50)   r>g-q6grq8<e-q6drq8>b-bq6<dq8a-q6gr2
(t50)   y126,1
(t50)   |:5r1:|@63y93,80v12r1q5>g^a-^q8aq5b^<c^1
(t50)   y127,0
(t50)   p3
(t50)   q8@63y93,80v12r*190g{a-ab-}b4r2r1e-{efg-}g4r*91q
8@49y93,00v13
(t50)   e-4c.gg16fe-gb-4a-r^2q4gga-a-abb
(t50)   <c*221
(t50)   @p120@31y93,60v14
(t50)   >f2..&@b0,-8192f&@b8191f1@b0e-2..&@b0,-2048e-&@b
-2048e-1
(t50)   @b0f1e-4v12@31y93,60p3
(t50)   y127,0
(t50)   r2.
(t50)   r>g-q6grq8<e-q6drq8>b-bq6<dq8a-q6gr2
(t50)   y126,1
(t50)   |:5r1:|@63y93,80v12r1q5>g^a-^q8aq5b^<c^1
(t50)   q5e-^d^q8cq5>b^<c^1q5v14g^r4.q8abq5<c^r
(t50)   q8v15c1^^q5>c^
(t70)   r*6
(t70)   @p120v12u110@10r4
(t70)   r4r1
(t70)   r2124<<c>bb-aa-gg-fee-dd-
(t70)   de-efg-g18u40
(t70)   gfrfe-r
(t70)   d
(t70)   {de-e}
(t70)   fr^>b<df
(t70)   a-ra-b-ra-gfe-
(t70)   {e-ef}
(t70)   g
(t70)   r^ce-g
(t70)   <c
(t70)   >e-
(t70)   g<c>b
(t70)   e-
(t70)   gb
(t70)   u110124e-efg-ga-18u40
(t70)   a-r^a-gfgrfre-drc
(t70)   r1
(t70)   v12u110@p8@73
(t70)   r1r1r1q8116r4.<ga-fge-fde-cde-efg-g8r8^2|:3r1:|
(t70)   @10@p120u90|:7r1:|r4{>dfgb<dg}4u-10
(t70)   {>dfgb<dg}4u-4{>dfgb<dg}4u-6
```

```
    557: (t70)    {>dfgb<dg}4r2.u|:7r1:|
    558: (t70)    r*1536
    559: (t70)    l16>>a-b-<cde-fga->ga-b-<cde-fg
    560: (t70)    >fa-<cfe-fde->e-a-<ce-de-cdr1r1
    561: (t70)    @10@p120u90|:7r1:|r4{dfgb<dg}4u-10
    562: (t70)    {>dfgb<dg}4u-4{>dfgb<dg}4u-6
    563: (t70)    {>dfgb<dg}4r2.u|:7r1:|
    564: (t70)    l8|:7r1:|r2^u70>g<ce-
    565: (t70)    grgfrfe-rd{de-e}fr^>b<dfa-ra-b-ra-gfe-{e-ef}gr^c
e-g
    566: (t70)    <c>e-g<c>be-gbb-ra-r^a-gfgra-rab^c^1
    567: (t70)    r*1536
    568: (t70)    r*1536
    569: (t70)    y126,1
    570: (t70)    u80e-^d^c>b^<c^1
    571: (t71)    r*6
    572: (t71)    y126,1
    573: (t71)    @p126l8v13u110@14y91,30y93,00r4
    574: (t71)    r4r1
    575: (t71)    r2^g<ce-
    576: (t71)    gr
    577: (t71)    gfrfe-r
    578: (t71)    d
    579: (t71)    {de-e}
    580: (t71)    fr^>b<df
    581: (t71)    a-ra-b-ra-gfe-
    582: (t71)    {e-ef}
    583: (t71)    g
    584: (t71)    r^ce-g
    585: (t71)    <c
    586: (t71)    >e-
    587: (t71)    g<c>b
    588: (t71)    e-
    589: (t71)    gb
    590: (t71)    b-r
    591: (t71)    a-r^a-gfgrfre-drc
    592: (t71)    r1
    593: (t71)    |:gce-c:|d{de-e}fr^2|:f>b<d>b:|<e-{e-ef}gr^2
    594: (t71)    |:gce-c:|d{de-e}fr^2f>b<d>b<gf^e-^1
    595: (t71)    r*1536
    596: (t71)    @11y91,50y93,30v8|:6r1:|q8rg-g^<e-d^>b-b<da-g^2
    597: (t71)    @14y91,30y93,0v13
    598: (t71)    |:>gce-c:|d{de-e}fr^2f>b<d>b<gf^e-^{e-ef}gr^2
    599: (t71)    r1r1r1
    600: (t71)    r1
    601: (t71)    r*768
    602: (t71)    r*1536
    603: (t71)    @11y91,50y93,30v8|:6r1:|q8rg-g^<e-d^>b-b<da-g^2
    604: (t71)    @14y91,30y93,0v13
    605: (t71)    |:>gce-c:|d{de-e}fr^2f>b<d>b<gf^e-^{e-ef}gr^2
    606: (t71)    r1r1r1
    607: (t71)    r2^>g<ce-
    608: (t71)    grgfrfe-rd{de-e}fr^>b<dfa-ra-b-ra-gfe-{e-ef}gr^c
e-g
    609: (t71)    <c>e-g<c>be-gbb-ra-r^a-gfgra-rab^c^1
    610: (t71)    @11y91,50y93,30v8|:6r1:|q8rg-g^<e-d^>b-b<da-g^2
    611: (t71)    @14y91,30y93,0v13
    612: (t71)    |:>gce-c:|d{de-e}fr^2f>b<d>b<gf^e-^{e-ef}gr^2
    613: (t71)    r1r1r1
    614: (t71)    r1
    615: (t71)    @73y91,70y93,30v9u110@p818
    616: (t71)    r1r2^<a-gfq5g^r2.r4
    617: (t71)    y127,0
    618: (t71)    q8<d1^{>gfe-}q5c^
    619: (t20)    r*6
    620: (t20)    @k5v13@718r4
    621: (t20)    (>ad)&(d>b-)
    622: (t20)    v12
    623: (t20)    l8g-q6gr2q8<e-d>b<da-g
    624: (t20)    r*94
    625: (t20)    @8v10
    626: (t20)    |:
    627: (t20)    q4l8<<c>g<c>g16&<cc16&>g
    628: (t20)    @b53|<c>g|:
    629: (t20)    bgbg16&bb16&gbg
    630: (t20)    :|:|q8{<c>ge-c}4
    631: (t20)    q4ge-cc16&gg16&e-c>b-<fffr^v14fe-d
    632: (t20)    e-rdrc>br
    633: (t20)    <cr^q8>b2.
    634: (t20)    @3v9p1
    635: (t20)    l16@q8|:|:4>ccc_4c~:||:8ddd_d~:||:4e-e-e-_e-~:|:
|
    636: (t20)    @8v9
    637: (t20)    q4l8<<<c>gcc16e-e-16dc>g<bf>bb16<ff16d>bg
    638: (t20)    <bf>bb16<bb16f>d>b<<<c>gcc16gg16e-c>g
    639: (t20)    <<c>cg<c>bq8{ce-fe-}4q4ca-cfc16ff16c>a-f
    640: (t20)    <gra-rabr<cr1
    641: (t20)    v12@7|:6r1:|
    642: (t20)    r>>>g-q6grq8<e-q6drq8>b-bq6<dq8a-q6gr2
    643: (t20)    v9@3p1l16@q8|:4>ccc_4c~:||:8ddd_d~:||:4e-e-e-_e-
~:|
    644: (t20)    |:4ccc_4c~:||:4ddd_d~:|r1r1
    645: (t20)    v7p1@q3l2@7<fe-dcv12
    646: (t20)    l8>g-q6gr2q8<e-d>b<da-g
    647: (t20)    r2
    648: (t20)    @8v9
    649: (t20)    q4l8<<c>gcc16e-e-16dc>g<bf>bb16<ff16d>bg
    650: (t20)    <bf>bb16<bb16f>d>b<<<c>gcc16gg16e-c>g
    651: (t20)    <<c>cg<c>bq8{ce-fe-}4q4ca-cfc16ff16c>a-f
    652: (t20)    <gra-rabr<cr1
    653: (t20)    v12@7|:6r1:|
    654: (t20)    r>>>g-q6grq8<e-q6drq8>b-bq6<dq8a-q6gr2
    655: (t20)    v9@3p1l16@q8|:4>ccc_4c~:||:8ddd_d~:||:4e-e-e-_e-
~:|
    656: (t20)    |:4ccc_4c~:||:4ddd_d~:|r1r1
    657: (t20)    @8v10
    658: (t20)    q4l8<<<c>g<c>g16&<cc16&>ge-cbgbg16&bb16&gd>b
    659: (t20)    <bfbf16&bb16&fd>b<<c>g<c>g16&<c>g16&e-c>g
    660: (t20)    <ge-cc16&gg16&e-c>b-<fff
    661: (t20)    v10
    662: (t20)    r^v14fe-de-rdrc>brq8
    663: (t20)    <c2..@7v10p1>>c4
    664: (t20)    v8l8f2..&(f>f)<<f1v7>e-2..&v8(e-c)&c1v7f1e-4r2.
    665: (t20)    v12@7
    666: (t20)    rg-q6grq8<e-q6drq8>b-bq6<dq8a-q6gr2
    667: (t20)    v9@3p1l16@q8|:4>ccc_4c~:||:8ddd_d~:||:4e-e-e-_e-
~:|
    668: (t20)    |:4ccc_4c~:||:4ddd_d~:|r1r1
    669: (t20)    @8v10p1l8r1r1r1r4<<<d1^^>c
    670: (t21)    r*6
    671: (t21)    @k5@1v10l8r4
    672: (t21)    r4r1
    673: (t21)    r2^>g<ce-
    674: (t21)    gr
    675: (t21)    gfrfe-r
    676: (t21)    d
    677: (t21)    {de-e}
    678: (t21)    fr^>b<df
    679: (t21)    a-ra-b-ra-gfe-
    680: (t21)    {e-ef}
    681: (t21)    g
    682: (t21)    r^ce-g
    683: (t21)    <c
    684: (t21)    >e-
    685: (t21)    g<c>b
    686: (t21)    e-
    687: (t21)    gb
    688: (t21)    b-r
    689: (t21)    a-r^a-gfgrfre-drc
    690: (t21)    r1
    691: (t21)    v11
    692: (t21)    |:gce-c:|d{de-e}fr^2|:f>b<d>b:|<e-{e-ef}gr^2
    693: (t21)    |:gce-c:|d{de-e}fr^2f>b<d>b<gf^e-^1
    694: (t21)    @7p1v9
    695: (t21)    q4l8>c>gcc16e-e-16dc>g<bf>bb16<ff16d>bg
    696: (t21)    <bf>bb16<bb16f>d>b<<<c>gcc16gg16e-c>g
    697: (t21)    <<c>cg<c>bq8{ce-fe-}4q4ca-cfc16ff16c>a-f
    698: (t21)    q8<gra-rabr<c^r2..
    699: (t21)    r*1536
    700: (t21)    @1v10
    701: (t21)    |:<gce-c:|d{de-e}fr^2f>b<d>b<gf^e-^{e-ef}gr^2
    702: (t21)    r1r1r1
    703: (t21)    r1
    704: (t21)    r*768
    705: (t21)    @7p1v9
    706: (t21)    q4l8>c>gcc16e-e-16dc>g<bf>bb16<ff16d>bg
    707: (t21)    <bf>bb16<bb16f>d>b<<<c>gcc16gg16e-c>g
    708: (t21)    <<c>cg<c>bq8{ce-fe-}4q4ca-cfc16ff16c>a-f
    709: (t21)    q8<gra-rabr<c^r2..
    710: (t21)    r*1536
    711: (t21)    @1v10
    712: (t21)    |:<gce-c:|d{de-e}fr^2f>b<d>b<gf^e-^{e-ef}gr^2
    713: (t21)    r1r1r1
    714: (t21)    r2^>g<ce-
    715: (t21)    grgfrfe-rd{de-e}fr^>b<dfa-ra-b-ra-gfe-{e-ef}gr^c
e-g
    716: (t21)    <c>e-g<c>be-gbb-ra-r^a-gfgra-rab^c^1
    717: (t21)    @7p2v3l8>>f2..&(f>f)<<f1>e-2..&(e-c)&c1f1e-4r2.r
1r1
    718: (t21)    @1v10
    719: (t21)    |:<<gce-c:|d{de-e}fr^2f>b<d>b<gf^e-^{e-ef}gr^2
    720: (t21)    r1r1r1
    721: (t21)    r1
    722: (t21)    @8v3p2l8r1r1r1r4a1^^>c
    723: (t22)    r*4
    724: (t22)    @k5v9@8l8r4
    725: (t22)    r*432
    726: (t22)    q4|:
    727: (t22)    e-ce-c16&e-e-16&c
    728: (t22)    @b53|e-c
    729: (t22)    q4l8d>b<d>b16&<dd16&>b<d>b<fdfd16&ff16&dfd
    730: (t22)    :|q8{e-c>ge-}4q4c>ge-e-16&<cc16&>gc>e-
    731: (t22)    <dddr^2v13gga-a-abb
    732: (t22)    <cr^q8g2.
    733: (t22)    @7v12
    734: (t22)    |:7r1:|rq8g-q4gq8bq4<c_5q8bq4<cq8d
    735: (t22)    r*1536
    736: (t22)    |:7r1:|@10p1v7r2^q8(<<d>>d)v5(<a->>a-)v4(<<d>>d)
    737: (t22)    r*1536
    738: (t22)    r*768
    739: (t22)    r*1536
    740: (t22)    |:7r1:|@10p1v7r2^q8(<<<d>>d)v5(<a->>a-)v4(<<d>>d
)
    741: (t22)    r*1536
    742: (t22)    @8v9
    743: (t22)    q4l8e-ce-c16&e-e-16&c>ge-d>b<d>b16&<dd16&>bgd
    744: (t22)    <fdfd16&ff16&d>bf<e-ce-c16&e-c16&>ge-c
    745: (t22)    <c>ge-e-16&<cc16&>gc>e-<ddd
    746: (t22)    @8
    747: (t22)    r^2v13gga-a-abbq8<c*211
    748: (t22)    @10p2v12|:7r1:|r4l16|:4(<<<<e->e-)&(e-<e-)&:|
    749: (t22)    ¥28|:(e->e-)&(e-<e-)&:|
    750: (t22)    (e-4>e-)¥0r2.|:7r1:|
    751: (t22)    @7v8p3l8r1r1r1r4>>>c1^^>c
    752: (t40)    r*6
    753: (t40)    @37v15u110@k3l8r4
    754: (t40)    r4
    755: (t40)    l8>>g-q6gr2q8<e-d>b<da-g
    756: (t40)    r2
    757: (t40)    @38y91,60y93,0
    758: (t40)    y126,1
    759: (t40)    q8>>cq2g<cq8>gcq2e-q8{<c>g}c>bq2<gbq8g>bq2<dq8{b
g}>b
    760: (t40)    bq2<gbq8g>bq2<dg-q8b-<cq2ccq8c>cq2e-q8{<c>g}c
    761: (t40)    |:cq2ccq8e-:|fq2ffr^2q4>gga-a-abb<cr^q8>g2.
    762: (t40)    |:
    763: (t40)    |:q8<<cq4cq8cq4cq8{c>g}q4e-q8<c>g|<dq4dq8dq4dq8{
d>a-}q4fq8<d>b
    764: (t40)    q8<dq4dq8dq4dq8>gq4gq8a-b:|
    765: (t40)    :|
```

```
   766: (t40)   |:q8<cq4cq8cq4cq8{c>g}q4e-q8<c>g|<dq4dq8dq4dq8{d
>a-}q4fq8<d>b
   767: (t40)   q8<dq4dq8dq4dq8>gq4gq8a-b:|
   768: (t40)   q8<cq4cq8cq4cq8{c>g}q4e-q8<ce-fq4fq8fq4fq8{>a-g}
q4fq8a-f
   769: (t40)   gra-rabr<c^q4>cccccc
   770: (t40)   @37
   771: (t40)   q8c
   772: (t40)   |:3f.q4<f.q8@38{ce-}@37:|>fa-b-<c|:>c.q4<c.q8{>g
f}ce-fg:|
   773: (t40)   |:f.q4<f.q8@38|{ce-}@37:|@37{fg}a-^rq4a-q8{a-gf}
2
   774: (t40)   r
   775: (t40)   y127,0
   776: (t40)   g-q6grq8<e-dr>b-b<da-gr2@38
   777: (t40)   y126,1
   778: (t40)   |:q8>cq4cq8cq4cq8{c>g}q4e-q8<c>g|<dq4dq8dq4dq8{d
>a-}q4fq8<d>b
   779: (t40)   q8<dq4dq8dq4dq8>gq4gq8a-b:|
   780: (t40)   q8<cq4cq8cq4cq8{c>g}q4e-q8<ce-fq4fq8fq4fq8{>a-g}
q4fq8a-f
   781: (t40)   gra-rabr<c^q4>cccccc
   782: (t40)   q8c
   783: (t40)   r1@37
   784: (t40)   y127,0
   785: (t40)   r1
   786: (t40)   l8<g-q6gr2q8<e-d>b<da-g
   787: (t40)   r2
   788: (t40)   y126,1
   789: (t40)   @38
   790: (t40)   |:q8>cq4cq8cq4cq8{c>g}q4e-q8<c>g|<dq4dq8dq4dq8{d
>a-}q4fq8<d>b
   791: (t40)   q8<dq4dq8dq4dq8>gq4gq8a-b:|
   792: (t40)   q8<cq4cq8cq4cq8{c>g}q4e-q8<ce-fq4fq8fq4fq8{>a-g}
q4fq8a-f
   793: (t40)   gra-rabr<c^q4>cccccc
   794: (t40)   @37
   795: (t40)   q8c
   796: (t40)   |:3f.q4<f.q8@38{ce-}@37:|>fa-b-<c|:>c.q4<c.q8{>g
f}ce-fg:|
   797: (t40)   |:f.q4<f.q8@38|{ce-}@37:|@37{fg}a-^rq4a-q8{a-gf}
2
   798: (t40)   r
   799: (t40)   y127,0
   800: (t40)   g-q6grq8<e-dr>b-b<da-gr2@38
   801: (t40)   y126,1
   802: (t40)   |:q8>cq4cq8cq4cq8{c>g}q4e-q8<c>g|<dq4dq8dq4dq8{d
>a-}q4fq8<d>b
   803: (t40)   q8<dq4dq8dq4dq8>gq4gq8a-b:|
   804: (t40)   q8<cq4cq8cq4cq8{c>g}q4e-q8<ce-fq4fq8fq4fq8{>a-g}
q4fq8a-f
   805: (t40)   gra-rabr<c^q4>cccccc
   806: (t40)   q8c
   807: (t40)   @38
   808: (t40)   |:q8{<cc}q4c|@37g@38q8c:|@37e-g
   809: (t40)   @38|:q8{>bb}q4b|@37<gq8@38>b:|q8@37<dg@38
   810: (t40)   |:q8{>bb}q4|b@37<f@38q8>b:|@37<dfg@38
   811: (t40)   {cc}q4c@37g@38q8c{cc}>b-{<c>b-a-g}4
   812: (t40)   {<cccc}4q4c>gq8@37{<cc}@38e-@37fg@37{ffff}4q4f@3
8cq8@37{ff}@38c>a-f
   813: (t40)   @37{<gce-g}4q4@38fdq8@37{e-e-}q4d@38>bq8<c^q4>cc
ccccq8c
   814: (t40)   v16<f1&f1c1&c1f1e-4r2.v15
   815: (t40)   r
   816: (t40)   y127,0
   817: (t40)   g-q6grq8<e-dr>b-b<da-gr2@38
   818: (t40)   y126,1
   819: (t40)   |:q8>cq4cq8cq4cq8{c>g}q4e-q8<c>g|<dq4dq8dq4dq8{d
>a-}q4fq8<d>b
   820: (t40)   q8<dq4dq8dq4dq8>gq4gq8a-b:|
   821: (t40)   q8<cq4cq8cq4cq8{c>g}q4e-q8<ce-fq4fq8fq4fq8{>a-g}
q4fq8a-f
   822: (t40)   gra-rabr<c^q4>cccccc
   823: (t40)   @37
   824: (t40)   q8c
   825: (t40)   @38q8<e-rdrc>br<c^q4>cccccccq8gr2ab<cr^c1^^>c
   826: (t41)   r*6
   827: (t41)   @9v13q1@h0@s255@m-500018r4
   828: (t41)   r*432
   829: (t41)   r*1536
   830: (t41)   |:|:8<<<<c:||:16d:||:8c:|:|
   831: (t41)   |:8c:||:16d:||:8c:||:8c:||:8d:||:8g:|r1
   832: (t41)   |:3f.f.r:|ffff|:g.g.q2{gg}q1ggq2{gg}q1g:|
   833: (t41)   |:f.f.|r:|q2{ff}q1fr2..r1r1
   834: (t41)   |:8c:||:16d:||:8c:||:8c:||:8d:||:8g:|r1
   835: (t41)   r*768
   836: (t41)   |:8c:||:16d:||:8c:||:8c:||:8d:||:8g:|r1
   837: (t41)   |:3f.f.r:|ffff|:g.g.q2{gg}q1ggq2{gg}q1g:|
   838: (t41)   |:f.f.|r:|q2{ff}q1fr2..r1r1
   839: (t41)   |:8c:||:16d:||:8c:||:8c:||:8d:||:8g:|r1
   840: (t41)   |:8{gg}g|gg:|{gggg}4{gggg}4gg{gg}ggg
   841: (t41)   {ffff}4ff{ff}fff{gggg}4ff{gg}f>g<fr1
   842: (t41)   r*1536
   843: (t41)   |:8c:||:16d:||:8c:||:8c:||:8d:||:8g:|r1
   844: (t1)    r*6
   845: (t1)    v16u110@1
   846: (t1)    @y26,50,110@y28,50,010@y29,50,60
   847: (t1)    @y26,48,110@y28,48,054@y29,48,80
   848: (t1)    @y26,45,115@y28,45,073@y29,45,100
   849: (t1)    @y26,41,127@y28,41,117@y29,41,110
   850: (t1)    @y28,57,120
   851: (t1)    @y28,49,120
   852: (t1)    @y24,39,70@y29,39,127@y30,39,127
   853: (t1)    @y29,54,127
   854: (t1)    @r118r4
   855: (t1)    r4>>ccr2.ddddr2
   856: (t1)    |:22c4:|r4.|:9c:|r4c4c2
   857: (t1)    |:16c2:|
   858: (t1)    |:14c2:|r|:7c:|
   859: (t1)    |:10c2:|c1c2c2{dddd}2r4c4
   860: (t1)    |:12c2:|c^c^cc^
   861: (t1)    c^|:7c:|
   862: (t1)    |:7c*0@u-65f16@u+65|:3@u-65f16@u+65:|:|c*0@u-65f
16@u+65@u-65f16@u+65rccr4^16|:7@u-65f16@u+65:|ddddr2
   863: (t1)    |:14c2:|r|:7c:|
   864: (t1)    |:10c2:|c1c2c2{dddd}2r4c4
   865: (t1)    |:12c2:|c^c^cc^
   866: (t1)    c^|:7c:|
   867: (t1)    |:12c2:|c^c^cc^
   868: (t1)    c^|:7c:|
   869: (t1)    |:3c1|r1:|c2{ccc}2c2c2{dddd}2r4c4
   870: (t1)    |:12c2:|c^c^cc^
   871: (t1)    cc2c2
   872: (t1)    c^c^cc^c^|:7c:|c2^ccc^4c2.r2c4
   873: (t2)    r*6
   874: (t2)    @r1v16u120r4
   875: (t2)    r4^4l16>>aaaaf2l8aaaa^l16<dddc>af
   876: (t2)    l8|:5r1:|r2..
   877: (t2)    u90
   878: (t2)    |:9f
   879: (t2)    u+5
   880: (t2)    :|
   881: (t2)    u120
   882: (t2)    r1
   883: (t2)    l16|:4r1|r4.aaf2:|r4.<ddddc>f<dc>fa
   884: (t2)    l16|:3r1r4.aaf2:|l8r2^<dc>f^fffl16ff<ddcc>af
   885: (t2)    r*648
   886: (t2)    <ddddc>a<dc>far1
   887: (t2)    r8{<dc>a}8<c4l24
   888: (t2)    d48c.^^c48>a.^^a48f.^^l16r2.<dc>af{ffff}2r4f4
   889: (t2)    |:3r1r4.aa|f2:|l8f<dc>ff^f^ff^f^fffl16ff<ddcc>af
   890: (t2)    r1r2^8<ddcc>aar4l16aaaaf2l8aaaa^2
   891: (t2)    l16|:3r1r4.aaf2:|l8r2^<dc>f^fffl16ff<ddcc>af
   892: (t2)    r*648
   893: (t2)    <ddddc>a<dc>far1
   894: (t2)    r8{<dc>a}8<c4l24
   895: (t2)    d48c.^^c48>a.^^a48f.^^l16r2.<dc>af{ffff}2r4f4
   896: (t2)    |:3r1r4.aa|f2:|l8f<dc>ff^f^ff^f^fffl16ff<ddcc>af
   897: (t2)    r1r1r1r2.<dc>afr1r2l8r<dc>ff^f^ff^f^fffl16ff<ddc
c>af
   898: (t2)    r*648
   899: (t2)    l8{<dd}dc>aff1
   900: (t2)    r8{<dc>a}8<c4l24
   901: (t2)    d48c.^^c48>a.^^a48f.^^l16r2.<dc>af{ffff}2r4f4
   902: (t2)    |:3r1r4.aa|f2:|l8f<dc>ff^f^ff^f^<d>al16<ddcc>ff<
dc>af
   903: (t2)    l8f^f^ff^f^fffl16ff<ddcc>afl8f2^fffr4f1^{<dc>a}f
4
   904: (t6)    r*6
   905: (t6)    @2p2@h0@m-1500@s255@k-1r4
   906: (t6)    r4^4l16v09>av09av09av09av11g2l8v09av09av09av09a^
l16v01<ev01ev01ev05cv09>av11g
   907: (t6)    l8|:5r1:|r2..
   908: (t6)    |:9v11g
   909: (t6)    :|
   910: (t6)    r1
   911: (t6)    l16|:4r1|r4.v09av09av11g2:|r4.v01<ev01ev01ev01ev
05cv11>gv01<ev05cv11>gv09a
   912: (t6)    l16|:3r1r4.v09av09av11g2:|l8r2^v01<ev05cv11>g^v1
1gv11gv11gl16v11gv11gv01<ev01ev05cv05cv09>av11g
   913: (t6)    r*648
   914: (t6)    v01<ev01ev01ev01ev05cv09>av01<ev05cv11>gv09ar1
   915: (t6)    r8{v01<ev05cv09>a}8v05<c4l24
   916: (t6)    v01e48v05c.^^v05c48v09>a.^^v09a48v11g.^^l16r2.v0
1<ev05cv09>av11g{v11gv11gv11gv11g}2r4v11g4
   917: (t6)    |:3r1r4.v09av09a|v11g2:|l8v11gv01<ev05cv11>gv11g
^v11g^v11gv11g^v11g^v11gv11gv11gl16v11gv11gv01<ev01ev05cv05cv09>
av11g
   918: (t6)    r1r2^8v01<ev01ev05cv05cv09>av09ar4l16v09av09av09
av09av11g2l8v09av09av09av09a^2
   919: (t6)    l16|:3r1r4.v09av09av11g2:|l8r2^v01<ev05cv11>g^v1
1gv11gv11gl16v11gv11gv01<ev01ev05cv05cv09>av11g
   920: (t6)    r*648
   921: (t6)    v01<ev01ev01ev01ev05cv09>av01<ev05cv11>gv09ar1
   922: (t6)    r8{v01<ev05cv09>a}8v05<c4l24
   923: (t6)    v01e48v05c.^^v05c48v09>a.^^v09a48v11g.^^l16r2.v0
1<ev05cv09>av11g{v11gv11gv11gv11g}2r4v11g4
   924: (t6)    |:3r1r4.v09av09a|v11g2:|l8v11gv01<ev05cv11>gv11g
^v11g^v11gv11g^v11g^v11gv11gv11gl16v11gv11gv01<ev01ev05cv05cv09>
av11g
   925: (t6)    r1r1r1r2.v01<ev05cv09>av11gr1r2l8rv01<ev05cv11>g
v11g^v11g^v11gv11g^v11g^v11gv11gv11gl16v11gv11gv01<ev01ev05cv05c
v09>av11g
   926: (t6)    r*648
   927: (t6)    l8{v01<ev01e}v01ev05cv09>av11gv11g1
   928: (t6)    r8{v01<ev05cv09>a}8v05<c4l24
   929: (t6)    v01e48v05c.^^v05c48v09>a.^^v09a48v11g.^^l16r2.v0
1<ev05cv09>av11g{v11gv11gv11gv11g}2r4v11g4
   930: (t6)    |:3r1r4.v09av09a|v11g2:|l8v11gv01<ev05cv11>gv11g
^v11g^v11gv11g^v11g^v01<ev09>al16v01<ev01ev05cv05cv11>gv11gv01<e
v05cv09>av11g
   931: (t6)    l8v11g^v11g^v11gv11g^v11g^v11gv11gv11gl16v11gv11
gv01<ev01ev05cv05cv09>av11gl8v11g2^v11gv11gv11gr4v11g1^{v01<ev05
cv09>a}v11g4
   932: (t7)    r*6
   933: (t7)    @2p1@h0@m-1500@s255@k1r4
   934: (t7)    r4^4l16v06>av06av06av06av02g2l8v06av06av06av06a^
l16v10<ev10ev10ev08cv06>av02g
   935: (t7)    l8|:5r1:|r2..
   936: (t7)    |:9v02g
   937: (t7)    :|
   938: (t7)    r1
   939: (t7)    l16|:4r1|r4.v06av06av02g2:|r4.v10<ev10ev10ev10ev
08cv02>gv10<ev08cv02>gv06a
   940: (t7)    l16|:3r1r4.v06av06av02g2:|l8r2^v10<ev08cv02>g^v0
2gv02gv02gl16v02gv02gv10<ev10ev08cv08cv06>av02g
   941: (t7)    r*648
   942: (t7)    v10<ev10ev10ev10ev08cv06>av10<ev08cv02>gv06ar1
   943: (t7)    r8{v10<ev08cv06>a}8v08<c4l24
   944: (t7)    v10e48v08c.^^v08c48v06>a.^^v06a48v02g.^^l16r2.v1
```



▶第１志望の大学に推薦入試で合格しました。８月号の「響子inCGわ～るど」に勇気づけられ，自分の力を信じてがんばったのがこの結果を生んだのだと思います。本当にありがとうございました。
浅場　誠志(20)神奈川県

```
0<ev08cv06>av02g{v02gv02gv02gv02g}2r4v02g4
(t7)    |:3r1r4.v06av06a|v02g2:|18v02gv10<ev08cv02>gv02g
^v02g^v02gv02g^v02g^v02gv02gv02g116v02gv02gv10<ev10ev08cv08cv06>
av02g
(t7)    r1r2^8v10<ev10ev08cv08cv06>av06ar4116v06av06av06
av06av02g218v06av06av06av06a^2
(t7)    116|:3r1r4.v06av06av02g2:|18r2^v10<ev08cv02>g^v0
2gv02gv02g116v02gv02gv10<ev10ev08cv08cv06>av02g
(t7)    r*648
(t7)    v10<ev10ev10ev10ev08cv06>av10<ev08cv02>gv06ar1
(t7)    r8{v10<ev08cv06>a}8v08<c4124
(t7)    v10e48v08c.^^v08c48v06>a.^^v06a48v02g.^^116r2.v1
0<ev08cv06>av02g{v02gv02gv02gv02g}2r4v02g4
(t7)    |:3r1r4.v06av06a|v02g2:|18v02gv10<ev08cv02>gv02g
^v02g^v02gv02g^v02g^v02gv02gv02g116v02gv02gv10<ev10ev08cv08cv06>
av02g
(t7)    r1r1r1r2.v10<ev08cv06>av02gr1r218rv10<ev08cv02>g
v02g^v02g^v02gv02g^v02g^v02gv02gv02g116v02gv02gv10<ev10ev08cv08c
v06>av02g
(t7)    r*648
(t7)    18{v10<ev10e}v10ev08cv06>av02gv02g1
(t7)    r8{v10<ev08cv06>a}8v08<c4124
(t7)    v10e48v08c.^^v08c48v06>a.^^v06a48v02g.^^116r2.v1
0<ev08cv06>av02g{v02gv02gv02gv02g}2r4v02g4
(t7)    |:3r1r4.v06av06a|v02g2:|18v02gv10<ev08cv02>gv02g
^v02g^v02gv02g^v02g^v10<ev06>a116v10<ev10ev08cv08cv02>gv02gv10<e
v08cv06>av02g
(t7)    18v02g^v02g^v02gv02g^v02g^v02gv02gv02g116v02gv02
gv10<ev10ev08cv08cv06>av02g18v02g2^v02gv02gv02gr4v02g1^{v10<ev08
cv06>a}v02g4
(t3)    r*4
(t3)    @r1v16u5511r4
(t3)    r*432
(t3)    >arrrarar4a2.
(t3)    a|:7r:|
(t3)    arrrar
(t3)    r2..
(t3)    a8^
(t3)    arrrar2^3d-6ar
(t3)    arrrar
(t3)    r2..
(t3)    a8^
(t3)    arrr
(t3)    arrrar
(t3)    r2..
(t3)    a8^
(t3)    arrrar2^3d-6ar
(t3)    arrrar
(t3)    18u127g-^g-^g-g-^g-*0u5511
(t3)    a8^
(t3)    arrrar
(t3)    r2..
(t3)    a8^
(t3)    arrrar2^3d-6ar
(t3)    arrrar
(t3)    r2..
(t3)    a8^
(t3)    a1r1r1r1r2..
(t8)    r*6
(t8)    v718r4
(t8)    r4@5<<<<c@4cr2@5c@4c@5c@4c@5c@4c^2
(t8)    116
(t8)    @4crcc@5c^@4ccc@5c@4cccr@5c^@4c^
(t8)    cccrcr^2
(t8)    @4crcc@5c^@4ccc@5c@4cccr@5c^@4c^
(t8)    cccrcrcr^^cc@5c^
(t8)    @4crcc@5c^@4ccc@5c@4cccr@5c^@4c^
(t8)    @5c^@4c2c^@5c^@4c^ccc^@5c^@4c^c^c^@5c^@4c^8.^2c^
@5c^
(t8)    |:4|:@4c^cc|:5c^:||c^:|@5c^:|
(t8)    |:3|:|:@4c^cc:||:3c^:||c^:|@5c^:|
(t8)    @4c^ccc^c^c^@5c^@4c^c^r2^8c^cc@5c^
(t8)    |:5@4c^^c^^cc^^c^c^@5c^:|
(t8)    @4c2..@5c^@4c^cc^^c^^^c^cc@5c^@4c^c^@5c^@4c2^8
(t8)    |:3|:@4c^ccc4c^c^c^|c^:|@5c^:|
(t8)    @4c^c@5c@4c^c^c^c^c^
(t8)    @5c^@4c4.cc^^cc^^@5c^
(t8)    |:@4c^cc^^c4c^c^@5c^:|r2..@5c^@4c^c^@5c^@4c^^^@5
c^@4c^@5c^
(t8)    |:3|:|:@4c^cc:||:3c^:||c^:|@5c^:|
(t8)    @4c^ccc^c^c^@5c^@4c^c^r2^8c^cc@5c^
(t8)    |:5@4c^^c^^cc^^c^c^@5c^:|
(t8)    @4c2..@5c^@4c^cc^^c^^^c^cc@5c^@4c^c^@5c^@4c2^8
(t8)    |:3|:@4c^ccc4c^c^c^|c^:|@5c^:|
(t8)    @4c^c@5c@4c^c^c^c^c^
(t8)    @5c^@4c4.cc^^cc^^@5c^
(t8)    |:3|:@4c^ccc4ccc^|c4:|c^@5c^:|@4c^cc|:5c^:|@5c^@
4c2^^c4@5c^
(t8)    |:5@4c^^c^^c4c^c^@5c^:|@4c2..@5c^
(t8)    @4c^cc^^c4c^c^@5c^@4c^c^@5c^@4c2^^
(t8)    |:3|:@4c^ccc4c^c^c^|c^:|@5c^:|
(t8)    @4c^c@5c@4c^c^c^c^c^
(t8)    @5c^^2.@4cc@5c^
(t8)    @4c^cc@5c^@4c^@5c^@4c4c^r2..@5c^@4c1r1r2c4
(t4)    r*6
(t4)    @r1v16u12718r4
(t4)    r4r1r2.>>e-4
(t4)    14u80|:11re-:|r2r1r2.e-
(t4)    14r
(t4)    |:16e-|r:|
(t4)    14r
(t4)    |:14e-|r:|r2{e-e-}8r4.
(t4)    |:11re-:|r2re-re-{e-e-e-e-}2re-
(t4)    14r|:12e-|r:|18
(t4)    e-re-re-e-re-^r^^
(t4)    {e-e-}r^^
(t4)    u100|:r2.e-4:|r1r2.u127e-4u80
(t4)    14r
(t4)    |:14e-|r:|r2{e-e-}8r4.
(t4)    |:11re-:|r2re-re-{e-e-e-e-}2re-
(t4)    14r|:12e-|r:|18
(t4)    e-re-re-e-re-^r^^
(t4)    {e-e-}r^^
(t4)    14r|:12e-|r:|18
(t4)    u100
(t4)    e-re-re-e-re-^r^^
(t4)    u80
(t4)    {e-e-}r^^
(t4)    14r1r1r1r1r1re-2^12e-6r4e-4r4e-4{e-e-e-e-}2ru127
e-u80
(t4)    14r|:12e-|r:|18e-re-re-e-re-14re-re-
(t4)    18u100e-re-re-e-re-^r^^u80{e-e-}r^^^2^e-e-e-r4e-
4r1r4
(t9)    r*6
(t9)    v1618r4
(t9)    r4r1r2.@6<<<e4
(t9)    14v11|:11r@6e:|r2r1r2.@6e
(t9)    14r
(t9)    |:16@6e|r:|
(t9)    14r
(t9)    |:14@6e|r:|r2{@6e@6e}8r4.
(t9)    |:11r@6e:|r2r@6er@6e{@6e@6e@6e@6e}2r@6e
(t9)    14r|:12@6e|r:|18
(t9)    @6er@6er@6e@6er@6e^r^^
(t9)    {@6e@6e}r^^
(t9)    v13|:r2.@6e4:|r1r2.v16@6e4v11
(t9)    14r
(t9)    |:14@6e|r:|r2{@6e@6e}8r4.
(t9)    |:11r@6e:|r2r@6er@6e{@6e@6e@6e@6e}2r@6e
(t9)    14r|:12@6e|r:|18
(t9)    @6er@6er@6e@6er@6e^r^^
(t9)    {@6e@6e}r^^
(t9)    14r|:12@6e|r:|18
(t9)    v13
(t9)    @6er@6er@6e@6er@6e^r^^
(t9)    v11
(t9)    {@6e@6e}r^^
(t9)    14r1r1r1r1r1r@6e2^12@6e6r4@6e4r4@6e4{@6e@6e@6e@6
e}2rv16@6ev11
(t9)    14r|:12@6e|r:|18@6er@6er@6e@6er@6e14r@6er@6e
(t9)    18v13@6er@6er@6e@6er@6e^r^^v11{@6e@6e}r^^^2^@6e@
6e@6er4@6e4r1r4
(p)
```

## リスト2　チャコの海岸物語

日本音楽著作権協会(出)許諾第9272282-201号

```
┌ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┐
| ◎Z－MUSIC（PCM8．X）                            ├ ┐
├ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┤ |
|                                                  | |
|                                                  | |
|        チャコの海岸物語                           | |
|      ━━━━━━━━━━━━━━━━━━━━━━━━                    | |
|        作詞／作曲　桑田佳祐                       | |
|      ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─                      | |
|        /Program 1992 August by ＫＡＭＥ           | |
|                                                  | |
|                                                  | |
|                          /Ramg-Network ID:443    | |
└ ┬ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┘ |
  └ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┘

*******************************************1992-8-11
.comment  チャコの海岸物語
          ━━━━━━━━━━━━━━━━━━━━━━━━━
.comment  Ｓouthern Ａll Ｓtars    Programmed by Ｋame
*****************************************************

/ Track Setup -------------------------

(i)(b0)(o131)

(m1,2000)(a1,1)
(m2,2000)(a2,2)
(m3,2000)(a3,3)
(m4,2000)(a4,4)
(m5,2000)(a5,5)
(m6,2000)(a6,6)
(m7,2000)(a7,7)
(m8,2000)(a8,8)
(m09,2000)(aAdpcm,09)
(m10,2000)(aAdpcm,10)
(m11,2000)(aAdpcm,11)
(m12,2000)(aAdpcm,12)
(m13,2000)(aAdpcm,13)
(m14,2000)(aAdpcm,14)
(m15,2000)(aAdpcm,15)
(m16,2000)(aAdpcm,16)

/ ADPCM SET ---------------------------

.adpcm_block_data=chako.zpd

/ SOUND SET ---------------------------

/-----< Ｓax >-----
/         AM  OM  WF  SY  SP PMD AMD PMS AMS PAN
(v1,0,    58, 15,  2,  1,  0,  0,  0,  0,  0,  3,  0
/         AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME
          18,  4,  0,  9,  1, 28,  1,  1,  3,  0,  0
```

▶この歳にして，ようやく愛車を手にすることができる（予定）。実際に手もとにくるのが正月過ぎなので，ちょっと遅い「お年玉」って感じ。そのとき，たぶん外は雪道なのでせいぜいぶつけないようにしたい。　　棟方　正治(25)青森県

```
          18, 14,  2,  9,  5, 46,  1,  6,  0,  2,  0
          18,  0,  0,  9,  1, 38,  1,  1,  0,  0,  0
          15,  3,  2,  9,  1,  2,  2,  2,  0,  0,  1)

/-----< Brass >-----
/         AM  OM  WF  SY  SP PMD AMD PMS AMS PAN
(v3,0,    58, 15,  2,  1,205, 20,  0,  3,  0,  3, 0
/         AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME
          16,  4,  0,  3,  4, 30,  1,  1,  3,  0,  1
          16, 31,  0,  3,  0, 69,  0,  2,  3,  2,  1
          14, 31,  0,  3,  0, 48,  0,  1,  3,  0,  1
          18, 31,  0,  4,  0,  0,  1,  1,  3,  0,  1)

/-----< E.Bass >-----
/         AM  OM  WF  SY  SP PMD AMD PMS AMS PAN
(v5,0,    56, 15,  2,  0,  0,  0,  0,  0,  0,  3,  0
/         AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME
          31, 13,  0,  5,  5, 29,  1,  0,  0,  0,  1
          31, 12,  0,  7,  5, 30,  0,  0,  3,  0,  1
          31,  9,  0,  6, 15, 38,  0,  0,  0,  0,  1
          31, 31,  0,  4,  0,  0,  0,  1,  3,  0,  1)

/-----< Violin >-----
/         AM  OM  WF  SY  SP PMD AMD PMS AMS PAN
(v6,0,    58, 15,  2,  0,  0,  0,  0,  0,  0,  3,  0
/         AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME
          20,  2,  0,  4,  1, 33,  1,  1,  0,  0,  0
          25,  6,  0,  7,  3, 50,  1,  5,  7,  0,  0
          28,  3,  0,  5,  1, 48,  1,  1,  0,  0,  0
          12,  4,  0,  5,  0,  0,  1,  1,  4,  0,  1)

/ MML SET UP ------------------------------

(t1)    o3q7l16@v116@5@s4,5@h5,3@m-10@a1@k-9     r8
(t2)    o4@q1l8@v123@1@s7  @h24   @m18            r8
(t3)    o4@q1l8@v119@1@s7  @h24   @m18@k-5r16.    r8
(t4)    o4q6l8 @v110@3@s4  @h20   @m23            r8
(t5)    o4q6l8 @v111@3@s4  @h20   @m23  r16       r8
(t6)    o4q6l8 @v107@3@s4  @h20   @m23  r         r8
(t7)    o4q7l16@v112@6@s4,5@h20,10@m23@a2         r8
(t8)    o4q7l16@v107@6@s4,5@h20,10@m23@a2@k-4r8   r8
(t9)    o5q8l8 v9                                 r8
(t10)   o5q8l8 v5                       r         r8
(t11)   o7q7l16v9                                 r8
(t12)   o7q7l16v7                       r8        r8
(t13)   o1q8l16v9                                 r8
(t14)   o1q8l16v7                       r.        r8
(t15)   o-1@r1l8v9                                r8
(t16)   o-1@r1l4v9                                r8

***********************************************************

(t1)
r1d4r8d8g&a8.g&a8.>a8r4<d&d#&e8&q5g8q7e4>b4r8b8<e4r8e8
a4r8e8r8c8>a8a8<e8d8e8g#8e8r4>a8|:
a8r4<d&d#&e8&q5g8q7e4d8r4<c&c#&q5d8>a8q7d4g8r4d8r8>b8g4<
c8r4d&d#&q5e8>b8q7e4a8r4<d#&d#q5e8g8q7e4
d8r4<c&c#&q5d8>a8q7d4e8r4d8r8d8>b4a8r4a8<a8r8a8e8
[$]e8r4<c8c#8d4.>>g8r4<a8a#8b4.c8r4d&e8.d8c4>
a8r4a8r8a8<e8d#8d4r8d4.r8d8
>b4r8b8b4r8b8<e4.e4>b8<d8e8e8e4e8e8r8<e4>
a8e8>a4<g8d8e8c4d8c8a8a8g8e4
f8c8d4g8a8g8c8c8d8c8a8a8g8e4d4r8f8f4a4>
a8r4<c8c4e4>b4r8b8<e4r8e8[*]|1>a8<e8c8>b8a8r8<<e4>:|
a4r8e8r8e8a8e8d4r8d8g&a8.g&a8.>a8r4<d&d#&e8g8e4>
b4r8b8<e4r8e8
a4r8e8r8c8>a8a8<e8d8e8g#8e8r8>a8a8r4<d&d#&e8&q5g8q7e4
d8r4<c&c#&q5d8>a8q7d4g4r4de&q5d8>b8q7g4<
c8r4d&d#&q5e8>b8q7e4a8r4<g&g#&q5a8e8q7>a4<
d8r4<c&c#&q5d8>a8q7d4e8r4d8r8e8d8>b8a4r8a8<a4a4[D.S.]
[CODA]a4r8e8r8e8e4d4r8f4.a4>a8r4b4.<e8c8
>b4.b2b8<e8r8>c2<<e4>a8r4d8&q5e8d8q7c4>a8r4<a8^2r1

(t2)
r1r1r2..a8~2b4b4_g#8g#4a8a1r2r_e~a~b|:
<c_c4c4>ba~b_ag4_f4fga~g_dddd~g4_e^2re~a~b
<c_c4c4c>ba<~dd4d.r16_c>b<c>b_ed#e<~dc4>a^4_r4e4e4
[$]f2.r~g4ga~ba_gfd~eg^2_ra4aae~a_g4f^2r4f4
~f#2g#~a4b&b1&b4_rg#a4b4<~c4c4>bg_de4~g4_e4re4
~a4a4_gg4_e^2.rede4f^4rfe~a4<c4r>_a4[*]
~b2<~dc4>a&|1a2___rg~a~b:|
a2__r2r1r2..a8~b4b4_g#8g#4a8a1_|:8r1:|r2e4e4[D.S.]
[CODA]b4b4<dc4>a&a2.rede4f4.rfea4<c4rc4>
b1b4r4b4b4<e1r4.cr2r1

(t3)
r1r1r2..a8~2b4b4_g#8g#4a8a1r2r_e~a~b|:
<c_c4c4>ba~b_ag4_f4fga~g_dddd~g4_e^2re~a~b
<c_c4c4c>ba<~dd4d.r16_c>b<c>b_ed#e<~dc4>a^4_r4e4e4
[$]f2.r~g4ga~ba_gfd~eg^2_ra4aae~a_g4f^2r4f4
~f#2g#~a4b&b1&b4_rg#a4b4<~c4c4>bg_de4~g4_e4re4
~a4a4_gg4_e^2.rede4f^4rfe~a4<c4r>_a4[*]
~b2<~dc4>a&|1a2___rg~a~b:|
a2__r2r1r2..a8~b4b4_g#8g#4a8a1_|:8r1:|r2e4e4[D.S.]
[CODA]b4b4<dc4>a&a2.rede4f4.rfea4<c4rc4>
b1b4r4b4b4<e1r4.cr2r1

(t4)
r1'f1a<d''e1a<c'r1r2
@z106,+3,+3,+3,+3,+3,+3
'e2a<c'@z0r8@v114p1|:4'e8g#b':|r4.
|:r4a8r8b8<c4>f8&f1r4g8r8a8b4<c8&
c2>b2r1r1r1r4<a8>r8_a4g4
[$]a2.r8b4b8<c8d8c8>b8a8g8g8<c8^2>~r4r1r1
r1e1e8e4e8e8r4.
p2~6>a8e8>a4<g8d8e8c4d8c8a8a8g8e4
f8c8d4g8a8g8c8c8d8c8a8a8g8e4<_p1
f1e1d#2e2[*]|1e8e8|:4e16:|e8r4.:|
e2.r4f1e1r1r2e2r8|:4e8:|~7p3
e8a8b8<c8c4c8r8c8>b8<c8>a8g4f8r8f8g8a8g8d4d8r8g4e8&
e2r8e8a8b8<c8c4c8r8c8>b8a8<d8d4d8r8c8>b8<c8>
b8e8d#8e8<d8c4>a8a2r2_8p1
[d.s.]
[coda]e1r1r1@1f#1g#4r4g#4e4@3e1e8r2..r1

(t5)
|:6r1:|p1
|:r4a8r8b8<c4>f8&f1r4g8r8a8b4<c8&
c2>b2r1r1r1r8.<e8>r8._1a4g4
[$]a2.r8b4b8<c8d8c8>b8a8g8g8<c8^2>~r4
r1r1r2...~3a1g#8g#4g#8g#8r4._r16
p2~5>a8e8>a4<g8d8e8c4d8c8a8a8g8e4
f8c8d4g8a8g8c8c8d8c8a8a8g8e8.<_p1
~3a1a1f#2g#2[*]|1a8a8|:4a16:|a8r4.r16_3:|
a2.r4a1a1r1r2a2r8|:4g#:|r16~4p3
e8a8b8<c8c4c8r8c8>b8<c8>a8g4f8r8f8g8a8g8d4d8r8g4e8&
e2r8e8a8b8<c8c4c8r8c8>b8a8<d8d4d8r8c8>b8<c8>
b8e8d#8e8<d8c4>a8a2r2_8p1
[d.s.]
[coda]a1r1r1r16@1_3f#1g#4r4g#4e4~@3a1a8r2..r1

(t6)
|:6r1:|p1
|:r4a8r8b8<c4>f8&f1r4g8r8a8b4<c8&
c2>b2r1r1r1r8a8r8r8_e4e4
[$]f2.rg4gabagfdeg^2r4
r1r1r2..~8b1b8b4b8b8r4.r8_
p2~6>a8e8>a4<g8d8e8c4d8c8a8a8g8e4
f8c8d4g8a8g8c8c8d8c8a8a8g8e8<_p1
~8<d1c1>a2b2[*]|1<c8c8|:4c16:|c8r4.r8_7:|
c2.r4d1c1r1r2c2r8>|:4b8:|r8~1
e8a8b8<c8c4c8r8c8>b8<c8>a8g4f8r8f8g8a8g8d4d8r8g4e8&
e2r8e8a8b8<c8c4c8r8c8>b8a8<d8d4d8r8c8>b8<c8>
b8e8d#8e8<d8c4>a8a2r2_9p1
[d.s.]
[coda]c1r1r1r8@1_6f#1g#4r4g#4e4~@3c1c8r2..r1

(t7)
r8<e8f8g8f8e8d8c8{ddd}2{dc>b}2<{ccc}2{c>ba}2~8b2g#2a1r1_
|:|:4r1:|r4<ceg<ce8>r4.r1r1{ab<cdeg}4a8r8r2
[$]r4.c#8d8r4c#8d8r2..r4.d#8e8>r4g#8a8r2..|:fefgagaa:|
<d#2c4>a4r8e8a8b8<e8>b8a8b8|:8r1:|[*]
|1<c>b<cded#eg#a8r4.>:|
r8e8f8g8f8e8d8c8{ddd}2{dc>b}2<{ccc}2{c>ba}2b2g#2
a1r1<<e1f1g2>b2
<c4c>bcde2a2e4a4f2d2r1r8{a<ce}8a8r8r2>[d.s.]
[coda]r2<{c>ba}2f2<{edc}2c2>a2
r1r8{b<cd}8e8r8r2r8e8d8e8d8c8>a8<e8a8r2..r1

(t8)
r8<e8f8g8f8e8d8c8{ddd}2{dc>b}2<{ccc}2{c>ba}2~8b2g#2a1r1_
|:|:4r1:|r4<ceg<ce8>r4.r1r1{ab<cdeg}4a8r8r2
[$]r4.c#8d8r4c#8d8r2..r4.d#8e8>r4g#8a8r2..|:fefgagaa:|
<d#2c4>a4r8e8a8b8<e8>b8a8b8|:8r1:|[*]
|1<c>b<cded#eg#a8r4.>:|
r8e8f8g8f8e8d8c8{ddd}2{dc>b}2<{ccc}2{c>ba}2b2g#2
a1r1<<e1f1g2>b2
<c4c>bcde2a2e4a4f2d2r1r8{a<ce}8a8r8r2>[d.s.]
[coda]r2<{c>ba}2f2<{edc}2c2>a2
r1r8{b<cd}8e8r8r2r8e8d8e8d8c8>a8<e8a8r2..r1

(t9)
refgfedc{ddd}2{dc>b}2<{ccc}2{c>ba}2_8o7b2e2a1
r|:4e:|r4.
|:a1d1g1c#2e2a1d1e1a2~o4a4g4
[$]f1_o7g#1c#1a1d1
b1~o4reab<e>bab_o7ee4eer4.a2g2c#1
f4f4fg4gc#1d1a1[*]b2e2|1|:5a:|r4.:|~o4
r<efgfedc{ddd}2{dc>b}2<{ccc}2{c>ba}2_o7b2e2a1
r8|:4e:|r4.a1d1g1c#2e2a1d1e1a2~o4a4g4[d.s.]
[coda]o7b2e2a2~o5{c>ba}2f1_o7a1
b1ere2.~o5rededc>a<ear4_o7a2^8r1

(t10)
refgfedc{ddd}2{dc>b}2<{ccc}2{c>ba}2_16o7b2e2a1
r|:4e:|r4.
|:a1d1g1c#2e2a1d1e1a2~o4a4g4
[$]f1_o7g#1c#1a1d1
b1~o4reab<e>bab_o7ee4eer4.a2g2c#1
f4f4fg4gc#1d1a1[*]b2e2|1|:5a:|r4.:|~o4
r<efgfedc{ddd}2{dc>b}2<{ccc}2{c>ba}2_o7b2e2a1
r8|:4e:|r4.a1d1g1c#2e2a1d1e1a2~o4a4g4[d.s.]
[coda]o7b2e2a2~o5{c>ba}2f1_o7a1
b1ere2.~o5rededc>a<ear4_o7a2^8r1

(t11)
|:4r1:|{rc>b}2{age}2r8_8|:4e8:|_e8e8e8
|:27r1:|~c2r8d8c8>b8a#1~
|:15r1:|o0d1c4.e8^2d#2e2c2r2
|:3r1:|o7{rc>b}2{age}2r8_|:4e8:|_e8e8e8~~
|:22r1:|o0d#2e2c2.r4d1c2.r4r1r1r1o6a8r4a8r2r1

(t12)
|:4r1:|{rc>b}2{age}2r8_8|:4e8:|_e8e8e8
|:27r1:|~c2r8d8c8>b8a#1~
|:15r1:|o0d1c4.e8^2d#2e2c2r2
|:3r1:|o7{rc>b}2{age}2r8_|:4e8:|_e8e8e8~~
|:22r1:|o0d#2e2c2.r4d1c2.r4r1r1r1o6a8r4a8r2r1

(t13)
r8<e8f8g8f8e8d8c8{ddd}2{dc>b}2<{ccc}2{c>ba}2~8b2g#2a1r1_
|:|:4r1:|r4<ceg<ce8>r4.r1r1{ab<cdeg}4a4r2
[$]r4.c#8d8r4c#8d8r2..r4.d#8e8>r4g#8a8r2..|:fefgagaa:|
<d#2c4>a4r8e8a8b8<e8>b8a8b8|:8r1:|[*]
|1<c>b<cded#eg#a8r4.>:|
_r8e8f8g8f8e8d8c8{ddd}2{dc>b}2<{ccc}2{c>ba}2b2g#2
a1r1<<e1f1g2>b2
<c4c>bcde2a2e4a4f2d2r1r8{a<ce}8a8r8r2>[d.s.]
[coda]r2<{c>ba}2f2<{edc}2c2>a2
r1r8{b<cd}8e8r8r2r8e8d8e8d8c8>a8<e8a8r2..r1

(t14)
r8<e8f8g8f8e8d8c8{ddd}2{dc>b}2<{ccc}2{c>ba}2~8b2g#2a1r1_
|:|:4r1:|r4<ceg<ce8>r4.r1r1{ab<cdeg}4a4r2
```

▶ うちの学校の先生は，IBMと国民機しか知らず，X68000の話をすると「へえ，そのパソコン68,000円で買えるんだ」といっていた。まるで2年前の自分を見ているようだった。
佐々木　義教(18)埼玉県

```
[$]r4.c#8d8r4c#8d8r2..r4.d#8e8>r4g#8a8r2..|:fefgagaa:|
<d#2c4>a4r8e8a8b8<e8>b8a8b8|:8r1:|[*]
|1<c>b<cded#eg#a8r4.>:|
_r8e8f8g8f8e8d8c8{ddd}2{dc>b}2<{ccc}2{c>ba}2b2g#2
a1r1<<e1f1g2>b2
<c4c>bcde2a2e4a4f2d2r1r8{a<ce}8a8r8r2>[d.s.]
[coda]r2<{c>ba}2f2<{edc}2c2>a2
r1r8{b<cd}8e8r8r2r8e8d8e8d8c8>a8<e8a8r2..r1

(t15)
r2..f
b4.f|:7f4.f:|f2rg16g16g4
|:
|:16f4.f:|
[$]b4.f|:13f4.f:|f1|:b4.ff4.f:|
f4.fg4.gg4.ff4.f|:f4.fffr4:||:f4.f:|[*]|1f2rfg4:|
f4.ffaaab4.f|:7f4.f:|f2r{aaa}aab4.f|:13f4.f:|fa16a16rar2
[d.s.]
[coda]
f4.f4ff4b4.fff4.|:f4.fff4f:|f4b4r4.ff4rff4.frffg^2r1

(t16)
r2r8{ccc}8c
|:4rcrc:|r8c8c8c8c2|:
|:7rcrc:|r4c8r8c8c8c8r8[$]
|:7rcrc:|c8c8c16c16c8c16c8c16c8c8
|:rcrc:|rcr8c8c
r2r8{ccc}8c|:3rcrc:|[*]|1c8c8c16c16c16c16cc:|rcc2
|:4rcrc:|r8{ccc}8c8{ccc}8c2
|:7rcrc:||:c16c16r8:|c8c16c16c8c8[d.s.]
{coda}
rcc.c8|:3rcrc:|ccr8{ccc}8crcrcc.c8^2r1

(p)

*************************************************************
```

## リスト3 チャコの海岸物語用カウンタ表示

```
 1:00004398 00000000   2:00004398 00000000   3:000043AA 00000000   4:00004398 00000000
 5:000043A4 00000000   6:000043B0 00000000   7:00004398 00000000   8:000043B0 00000000
 9:00004398 00000000  10:000043B0 00000000  11:00004398 00000000  12:000043B0 00000000
13:00004398 00000000  14:000043AA 00000000  15:00004398 00000000  16:00004398 00000000
```

## リスト4 CHAKO.CNF

```
1=¥ADDITION¥piano.pcm,p12,v25
2=1,p12
.o0c=1,p-12
.o0d#=1,p-9
.o0e=1,p-8
.o0f=1,p-7
.o0f#=1,p-6
.o0g=1,p-5
.o0g#=1,p-4
.o0a=1,p-3
.o0b=1,p-1
.o1c=1
.o1d=1,p2
.o1e=1,p4
.o1f=1,p5
.o1g=1,p7
.o1a=1,p9
.o4f=.o1f,mo0f
.o4g=.o1g,mo0g
.o4g#=1,p8,mo0g#
.o4a=.o1a,mo0a
.o4b=1,p11,mo0b
.o5c=2,mo1c
.o5d=2,p2,mo1d
.o5e=2,p4,mo1e
.o5f=2,p5,mo1f
.o5g=2,p7,mo1g
.o5a=2,p9,mo1a
.erase 1
.erase 2
1=.o0e,mo0a
.o7a=1,mo1c
.erase 1
1=.o0f,mo0a
.o7d=1,mo1d
.erase 1
1=.o0d#,mo0f#
.o7b=1,mo0a
.erase 1
1=.o0e,mo0g#
.o7e=1,mo0b
.erase 1
1=.o0g,mo0b
.o7g#=1,mo1d
.erase 1
1=.o0c,mo0e
.o7c#=1,mo0g
.erase 1
.o7f=.o7c#,p5
.o7g=.o7c#,p7
.erase o0c
.erase o0d#
.erase o0e
.erase o0f
.erase o0f#
.erase o0g
.erase o0g#
.erase o0a
.erase o0b
.erase o1c
.erase o1d
.erase o1e
.erase o1f
.erase o1g
.erase o1a
1=¥ADDITION¥dsg_o2c.pcm,p12,v32
.o5a#=1,p-3
.o5b=1,p-1
.o6c=1
.o6d=1,p2
.o6e=1,p4
.o6f=1,p5
.o6g=1,p7
.o6a=1,p9
.o6b=1,p11
.o7c=1,p12
.erase 1
1=¥ADDITION¥str.pcm,p12,v20
.o1g#=1,p-4
.o1a=1,p-3
.o1b=1,p-1
.o2c=1
.o2d=1,p2
.o2e=1,p4
.o2f=1,p5
.o2g=1,p7
.o2g#=1,p8
.o2a=1,p9
.o2b=1,p11
.o3c=.o2c,p12
.o3c#=.o3c,p1
.o3d=.o3c,p2
.o3d#=.o3c,p3
.o3e=.o3c,p4
.o3f=.o3c,p5
.o3g=.o3c,p7
.o3g#=.o3c,p8
.o3a=.o3c,p9
.erase 1
1=¥ADDITION¥cho.pcm,v16
.o0c=1
.o0d=1,p2
.o0d#=1,p3
.o0e=1,p4
.erase 1
.o-1c=¥SNARE¥snare3_.pcm,v57
/.o-1c=¥SNARE¥wdsd1.pcm,v57/←原曲はこちらの方が近い
.o-1f=¥BASS¥kick1.pcm,v29
.o-1g=¥CYMBAL¥crsh0.pcm,v14
.o-1a=¥TOMTOM¥romt2.pcm,v34
.o-1b=.o-1g,mo-1f
```

## リスト5 LAST CHRISTMAS(先月号に掲載)の音色データ

(ファイルサイズは1694バイト)

```
0000  26 DF 2D 6C 68 35 2D 75  : DD
0008  06 00 00 4B 29 00 00 6D  : E7
0010  06 89 19 20 01 0D 4C 41  : 63
0018  53 54 5F 58 4D 41 53 2E  : 6D
0020  4D 44 44 EF 44 48 00 00  : 50
0028  04 25 6C 72 B6 37 22 FF  : 15
0030  27 95 51 48 8B EB 1E 66  : 4F
0038  66 66 38 91 C7 1B B5 2F  : 5B
0040  BF FF E6 36 E3 93 64 DD  : 91
0048  DD D9 24 E3 A8 D0 2B AA  : 0A
0050  B8 DD 0A 10 75 09 04 BC  : ED
0058  58 18 E2 84 82 E2 C1 05  : 00
0060  80 CB 00 96 0A 2C 01 38  : 50
0068  0D D7 A4 71 36 EF A8 9E  : 64
0070  C2 65 44 C8 93 1A 52 BC  : EE
0078  AA A9 3E 30 AF 91 69 D1  : 3B
-------------------------------------
SUM:  08 9D FA 15 2F 1C 79 90  F719

0080  A7 46 FC 0D 55 CF 03 16  : 33
0088  94 5E 4D AB 7F CD 3A 17  : 87
0090  C6 A8 26 1D C7 A4 8F 67  : 12
0098  D7 3F 39 EB D5 53 FE 8C  : EC
00A0  80 F1 60 D2 5D 4D 61 E4  : 92
00A8  FB 0F C1 CD 3F 99 E4 07  : 5B
00B0  E7 5A 64 2A 97 0F F1 E7  : 4D
00B8  F2 2F D3 A1 F1 21 F2 AB  : 44
00C0  49 92 3F CE DF FD 04 F8  : C0
00C8  EB 4F 76 3F 8C 28 84 49  : 70
00D0  96 93 6E C9 31 69 34 22  : 50
00D8  56 63 96 6C 93 83 17 DA  : C2
00E0  01 12 81 E7 7C D3 12 E0  : BC
00E8  98 2A 87 E7 F0 21 86 96  : 5D
00F0  D4 69 54 62 A9 86 59 0C  : 87
00F8  13 15 E2 B4 4C 78 E8 BF  : 29
-------------------------------------
SUM:  CC A5 F7 50 24 AC 9E 1B  D77F

0100  5E C3 57 6C 87 CB 20 ED  : 43
0108  40 89 71 21 F0 CF 09 2D  : 50
0110  69 84 F1 26 C8 06 64 E2  : 18
0118  8D C7 D6 92 04 FE 1E F2  : CE
0120  6C C4 09 AA 21 30 F9 B5  : E2
0128  39 6B F1 DE 58 E9 87 EF  : 2A
0130  C8 9B 0D 91 F6 94 36 1F  : E0
0138  9B DE 50 03 5E 05 0B 8A  : C4
0140  16 69 D0 EB 0A 1A D0 9B  : C9
0148  6B CC 9B 4C 95 77 7C 04  : AA
0150  DA FD C9 8B 7B 8F 88 4C  : 09
0158  06 E0 09 E4 1F E9 CF BC  : 66
0160  89 B7 44 C4 FA 8B 72 61  : A0
0168  4E DF C1 25 DE 9E F6 1A  : 9F
0170  9F BE 41 F7 F5 26 D5 CD  : 52
0178  CD 60 86 73 26 CB D6 8E  : 7B
-------------------------------------
SUM:  40 05 EF 5A 3C 73 22 B8  A448

0180  32 3B 1A FE F7 36 AF 6C  : CD
0188  E0 DB B7 A7 84 93 F8 B5  : DD
0190  40 80 D6 11 AA E6 E6 A4  : C1
0198  3B 0E 7F CD CD AE B4 39  : FD
01A0  B5 88 F4 7B 66 DC 25 58  : 6B
01A8  54 E7 B0 28 0D 45 24 0B  : 94
01B0  8B 54 D3 85 0D F8 A7 B5  : 98
01B8  60 A6 49 13 A6 D3 FD 01  : D9
01C0  A9 34 C1 0C C3 DB FB 39  : 7C
01C8  B6 04 EB B7 A6 03 89 5D  : EB
01D0  37 72 8F 8E D8 4E 72 3B  : 99
01D8  7A 18 BE 97 D4 08 6D 64  : 94
01E0  B5 13 0F F0 6A 78 26 3F  : 0E
01E8  D2 25 1E A4 D9 95 4E 09  : 7E
01F0  48 EA E8 4D A3 3D 17 E0  : 3E
01F8  4D 87 52 CB 26 DD 8F 6B  : EE
-------------------------------------
SUM:  AD 78 46 52 39 A4 AB DF  D313

0200  28 21 3E 84 D9 EC D3 A0  : 43
0208  10 FE 44 D9 B1 4F CF 12  : 0C
0210  6D 9A 63 AF D0 9B 48 C6  : 92
0218  47 CC 9B 6A 38 61 7C 89  : B6
0220  B1 99 B8 94 1B FA 93 6E  : AC
0228  D9 0F D0 9B 57 D4 9B 57  : 70
0230  30 28 23 2F E6 4D 93 06  : 76
0238  FD 85 08 24 0C A0 1C 0E  : 84
0240  E8 BD 3C 2F 0D DA D9 DC  : AC
0248  E6 D7 54 64 14 C1 F7 7C  : BD
0250  1E 59 43 49 84 0F 63 42  : 3B
0258  12 6F 79 31 57 F3 86 90  : 8B
0260  E1 73 06 D0 E0 F3 8F 74  : 00
0268  34 8D BB 18 38 04 23 4D  : 40
0270  B2 8C E5 6E C4 DF 75 D1  : 7A
0278  28 75 07 CD 88 A9 3D 3C  : 1B
-------------------------------------
SUM:  90 37 2C 28 56 0E 60 D2  B03C

0280  20 6D F5 49 34 46 C3 59  : 61
0288  8C E5 6D E3 2D E3 2E 0F  : 0E
0290  B1 36 3E 00 8C E1 A4 9E  : D4
0298  73 0C 83 6B 27 EB EF 89  : F7
02A0  67 B0 D4 FA 13 69 74 26  : FB
02A8  CD AA 70 A3 42 5D 09 B2  : E4
02B0  05 D9 3C A5 04 A0 31 0F  : A3
```

▶12月号ショートプロ大集合にあった「裸眼立体視(ランダムドット)」の出力例は，本当にこんなものが立体に見えるのだろうか，と半信半疑でした。しかし，実際に四角形が浮かび上がってきたときには感動してしまいました。
横田　紀明(25)山口県

```
02B8  47 36 A3 D9 19 54 C4 23  : 4D
02C0  5B C4 20 DC 7A F1 B7 DA  : 17
02C8  1F 63 E3 E4 84 14 20 C6  : C7
02D0  87 BE 71 18 B1 89 ED 59  : 4E
02D8  B6 34 07 53 D2 E4 9B B4  : 49
02E0  69 40 48 DB 79 5D 09 B3  : 5E
02E8  26 E3 36 6A 35 9C 64 E6  : C4
02F0  22 AA 36 E2 18 A3 54 C7  : BA
02F8  03 D0 E5 89 8A EC 90 6C  : B3
-------------------------------------
SUM:  BB B3 5A 8D 57 A9 A6 12  597E

0300  D3 04 44 B8 A6 22 30 D3  : 9E
0308  E0 65 7B 5D 66 F6 57 CE  : 9E
0310  2E 21 29 EF 44 C4 1F D7  : 65
0318  5B BD 30 04 DA 89 88 60  : 97
0320  9E 69 8E CE C3 53 F9 D8  : 4A
0328  61 D5 CD AD 59 FC 51 C8  : 1E
0330  DC C9 B2 03 E9 44 E1 A7  : 0F
0338  12 70 9D 1C AB 78 93 6A  : 5B
0340  53 90 57 25 ED 10 80 43  : 1F
0348  BE CD EE 6D AF 12 4E B4  : A9
0350  10 30 F8 21 B3 10 43 82  : E1
0358  40 F0 DE CB D1 CD A6 60  : 7D
0360  2C 41 C3 1A 68 D6 24 7E  : 2A
0368  B1 33 C2 E4 78 00 59 03  : 5E
0370  8F E0 DA 39 B5 5C C9 1B  : 77
0378  F2 73 6D D3 6D A8 10 90  : 5A
-------------------------------------
SUM:  E8 02 A9 2A FC 49 F9 8E  52F9

0380  A4 EC 0F D7 D8 20 F6 C8  : 2C
0388  73 26 D2 AB A1 36 72 93  : F2
0390  69 1B 46 B9 13 66 2C 29  : 51
0398  7B 0D CD A1 CE 7F 6B 9B  : 49
03A0  54 9B 62 E6 DB 76 C1 6B  : B4
03A8  C0 F8 72 73 6C E0 8A 77  : EA
03B0  CA B4 05 C9 07 9B 3B E9  : 12
03B8  A4 86 DF 55 4A D0 14 A7  : 33
03C0  9E 0C C5 7B 93 17 99 58  : 85
03C8  34 C5 B9 49 61 62 56 CF  : E3
03D0  A9 22 99 12 7E BB 07 94  : 4A
03D8  5C 7F ED E9 5A 49 89 0E  : EB
03E0  C3 2E 3C 86 50 A5 8D 86  : BB
03E8  06 79 46 58 3C 24 30 01  : AE
03F0  EA F8 4B 8D E7 BB 61 A4  : 61
03F8  9B 9B 66 BA 3C F2 06 E7  : 71
-------------------------------------
SUM:  A2 B3 E3 37 6D EF 3C 6C  58E8

0400  0D 6A F4 74 8A 29 FF E8  : 79
0408  A3 DE 05 16 BA CE EE 62  : 74
0410  1D D3 8F 31 97 8F 31 08  : 0F
0418  D3 55 A8 89 CC 43 A1 F2  : FB
0420  97 A7 AF 31 06 A7 27 31  : 23
0428  0D 87 1E 63 2C 39 88 E0  : E2
0430  17 CB 98 9C 92 4A 88 15  : 8F
0438  F9 93 68 2C F3 C1 06 BF  : 99
0440  A1 36 5A 02 5A 8E B4 01  : D0
0448  93 6B D2 B4 1D FB C9 B3  : 18
0450  71 26 C6 21 1F 8C 85 23  : D1
0458  7F C7 86 FF B8 69 2F 38  : 53
0460  46 B0 83 CA 75 A0 33 89  : 14
0468  59 11 55 74 F6 70 D2 42  : AD
0470  69 81 E4 1B AC FB 7D D5  : E2
0478  EE 86 91 52 1D 49 5A 96  : AD
-------------------------------------
SUM:  6E 52 C2 21 E0 86 09 6E  2A43

0480  D9 99 EC 5E C5 B5 31 33  : 9A
0488  69 8A 62 5C 13 0C A1 A4  : 15
0490  F0 4C 7B 38 2F F8 86 93  : 2F
0498  AB 9B 60 AE 6D D6 7D A7  : BB
04A0  CC 9B 27 7D 99 EA B8 F4  : 3A
04A8  B1 40 3C 3A D4 37 40 D8  : 8A
04B0  22 DC 09 B4 B7 39 B6 58  : B9
04B8  88 49 D5 D1 E4 E6 D8 EE  : 07
04C0  73 6B 98 90 C8 A4 7D E7  : D6
04C8  2A A8 B3 0C 66 DF 5A 43  : 73
04D0  9F 5C E9 FB AC B2 AE 41  : 2C
04D8  F8 90 CC 1C 3A 4D D5 47  : 13
04E0  36 8A 17 62 99 28 0F F0  : F9
04E8  D4 82 15 B8 8D 78 23 85  : D0
04F0  16 1A 48 4F 51 4A 69 21  : EC
04F8  98 E7 10 9F 7A 63 DD 67  : 4F
-------------------------------------
SUM:  F0 16 EE 97 81 9E 2D D2  CB28

0500  0E 3B E4 DB 6D F4 4C 76  : 2B
0508  2A 62 5C 13 05 8A 67 9A  : 8B
0510  62 5C 13 0A 36 A7 04 C4  : 80
0518  1A 1C 93 1E C3 53 E8 4D  : 32
0520  99 D6 80 94 94 42 DD EE  : 24
0528  B9 DC CB BC 9C DA 60 5B  : 4D
0530  C5 CD B9 C9 B7 FF B9 B5  : 38
0538  D7 F6 73 6F FD 90 10 F3  : 3F
0540  F4 26 CA E6 DB FD D7 3B  : B4
0548  96 7F 08 38 DC DA 20 62  : 8D
0550  32 66 2E 27 38 FE 64 4D  : D4
0558  95 B3 57 D2 9A 4E F7 36  : 86
0560  AE 02 14 8E D5 8C CE 47  : C8
0568  0C E8 09 21 70 0D FB 7A  : 10
0570  35 FA AD 5A 74 45 35 B0  : D4
0578  DC B8 61 41 66 F4 19 0F  : B8
-------------------------------------
SUM:  BE E4 DF FF F7 18 0E B2  6476

0580  33 A4 71 F7 DD 37 21 24  : 98
0588  96 68 4D B3 90 A9 6B 80  : 22
0590  88 BB 9C 8E CC D4 74 1D  : 9E
0598  05 0D C1 90 85 B4 26 D3  : 95
05A0  8C D7 BD 31 41 7C DC 2B  : 15
05A8  0D 85 18 6C 3E 29 8A B8  : BF
05B0  23 51 4D 30 B7 04 C0 2E  : 9A
05B8  9B 93 13 A6 81 59 B9 21  : 9B
05C0  65 65 3C 56 69 80 C8 24  : 31
05C8  4B 27 1D 1B 4F 52 5A DF  : 84
05D0  66 AC 49 32 DB AA 56 30  : 98
05D8  3E 4C E6 0B AF 24 C5 87  : 9A
05E0  B6 B9 2D CD 08 14 F6 D1  : 4C
05E8  84 79 DE E6 0B A5 B4 A7  : CC
05F0  2D 5A D1 7C 4E F2 BB 75  : 44
05F8  43 B5 90 BC D1 81 90 7C  : A2
-------------------------------------
SUM:  AB D9 44 D4 E9 36 37 E9  EBC7

0600  E4 30 5E B2 92 77 66 3C  : CF
0608  C4 FA 47 DE 9B E3 8E D9  : C8
0610  FE CC 4F 5E F6 4C F8 73  : 24
0618  15 2D 56 28 42 FB B9 89  : 3F
0620  14 3E 40 53 C3 98 B6 C9  : BF
0628  B6 D2 F7 F3 13 36 72 26  : 53
0630  CC 9C 1E 6C C2 CE E2 61  : C5
0638  88 C9 C1 1F F5 23 E3 A2  : CE
0640  3C 71 18 36 23 C9 11 F1  : E9
0648  D9 1E 64 8F 93 C8 65 CB  : 75
0650  E1 96 AE 9E 2E 77 AE 58  : 6E
0658  36 E8 8F A4 C8 E1 FC F5  : EB
0660  CB 1F F3 03 E4 8E 5B 6F  : 1C
0668  C3 2D 14 56 CF F9 B2 CF  : A3
0670  6C B2 7B 57 4F F1 F1 CB  : EC
0678  60 AD B3 D9 5B 8E 0A DD  : 69
-------------------------------------
SUM:  5F 50 4E 77 FB 4F BA F2  EFC4

0680  5F 8A B7 5D 3D 98 CF 6C  : 0D
0688  B5 94 56 EA F6 E1 29 55  : DE
0690  45 6D 33 7F 1C B2 68 B0  : 4A
0698  A7 E1 32 DF 50 00 00 00  : E9
06A0  00 00 00 00 00 00 00 00  : 00
06A8  00 00 00 00 00 00 00 00  : 00
06B0  00 00 00 00 00 00 00 00  : 00
06B8  00 00 00 00 00 00 00 00  : 00
06C0  00 00 00 00 00 00 00 00  : 00
06C8  00 00 00 00 00 00 00 00  : 00
06D0  00 00 00 00 00 00 00 00  : 00
06D8  00 00 00 00 00 00 00 00  : 00
06E0  00 00 00 00 00 00 00 00  : 00
06E8  00 00 00 00 00 00 00 00  : 00
06F0  00 00 00 00 00 00 00 00  : 00
06F8  00 00 00 00 00 00 00 00  : 00
-------------------------------------
SUM:  00 6C 72 A5 9F 2B 60 71  C4B1
```

## (善)のゲームミュージックでバビンチョ

セガのアーケードゲーム「シャドーダンサー」のエンディングをやっと見ることができた。ショー・コスギの忍者ものの映画の影響をうけてか，最後のボスはご丁寧に手をついてお辞儀をしちゃう。笑った。あと，Oh!X スタッフ内ではこの主人公忍者がしゃべる呪文についての議論が沸騰中。私には「おんはらうっさうなー，おりゃー」「ランバダはサンバじゃござらんばー，おりゃー」「おんさばやか，そばっ，おりゃー」と聞こえるんだけど，合ってる？

**●スタークルーザーII　　CD: POCH-2203**
**ポリドール　　2,500円(税込)　発売中**

PC-9801用に発売された3Dアクションアドベンチャーゲーム「スタークルーザーII」のオールアレンジBGM集。ゲームには未使用の曲までもアレンジされて収録されている。作曲，アレンジともにアルシスの吟遊詩人・山中季哉氏が率いるアルシスサウンドチームによるもので，作曲者たちからのメッセージが余すところなく伝わってくる。打ち込みによる作品がメインで，使用楽器もD70, SY77, R8, SC55といった一般的なものだから，DTMのお手本としても価値がある。各パートのエフェクタの割り振りとミキシングのバランスがイマイチのような感もあるが，ファン必携の一枚であることは否定しない。ゲームのほうもX68000への移植を期待したい。

お勧め度　8

**●コナミ・オールスターズ1993**
**CD: KICA-9016〜9018**
**キングレコード　6,200円(税込)　12/24発売**

毎年，濃厚な(?)内容でファンを楽しませてくれた恒例「コナミ・オールスターズ」の1993年版。今年もCD 3枚組の構成で，ディスク1は「夢のミュージックステーション」と題するボーカルアレンジバージョン集。矩形波倶楽部のメンバーやコナミ広報の早坂，砂井(誰だそれ)が歌ってるけど，ま，音楽は本人が楽しければいいという要素があるわけで……。ディスク2は「さゆ鈴のコナミレーベルインフォメーション番外編」という，丸ごと「さゆ鈴」のおしゃべりCD。声が妙に硬いのはなんとかならなかったのか。ディスク3は「アメコミコネクション」と題され，コナミ・アメコミゲームシリーズからセレクトされたオリジナルゲームサウンドが収録されている。今回はゲームミュージックのCDというよりはファンサービス的な色が濃い。悪くいえば「楽屋オチ」「内輪うけ」もので，残念ながら万人に勧められる内容ではない。

お勧め度　6

**●重装機兵ヴァルケン　　CD: TOCT-6861**
**東芝EMI　　2,500円(税込)　12/16発売**

昔から，NCSのゲームミュージックはそのゲーム機のオーバースペックを実現しているといわれていたが，この「重装機兵ヴァルケン」(スーパーファミコン)はその定説をさらに裏づける完成度といえる。アレンジバージョン3曲とオリジナルサウンドという構成だが，オリジナルサウンドもスーパーファミコン本体からの録音ではなく曲が開発された機材環境での演奏を収録しているため，音質レベルは高い。ゲームの性質上，状況描写的な曲が多いが，数曲あるダンサンブルタイプは抜群のセンスのよさで「踊れる」BGMに仕上がっている(トラック5)。

お勧め度　8

**●Falcom Neo Classic**
**From STUDIOS IN LONDON CITY**
**CD: KICA1114〜1115**
**キングレコード　4,500円(税込)　発売中**

ファルコムレーベル5周年記念として2枚組+αのフルアレンジアルバムが発売された。ディスク1には，あのロンドン交響楽団がファルコムのゲームミュージックを演奏したものを収録。そしてディスク2は松武秀樹氏らがファルコムの歴代のゲームミュージックをシンセサイザアレンジで蘇らせている。そして，初回特典でロンドン交響楽団のリハーサルの模様を収めたシングルCDが付いてくる。実際に聞いてみた感想としては，ディスク1は選曲がイマイチという感が否めないがアレンジは最高，耳当たりも良好完璧，ディスク2はリズムも面白いし音にも工夫がみられるが曲調に合っていないアレンジが目立った(わざとだという話もある)。

お勧め度　7

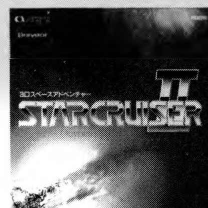
### 終わりに

えーと，ラジオ大阪で「ゲームミュージッククラブ」っていう番組が10月11日から放送が開始されてるんだって(OBC 1314kHz毎週日曜日PM8:30〜9:00)。パーソナリティは，コナミ・矩形波倶楽部のリーダーの古川もとあき氏。聞きたいけど関西地区にしか流れていないから私は無理だな，残念。番組へのお便りは

〒530-13　ラジオ大阪

「ゲームミュージッククラブ」係　まで。

ではまた来月。






CREATIVE COMPUTER MUSIC

Creative Computer Music入門(16)

# 金管楽器のDTM利用

オーケストレーションのアレンジは，ストリングスに続いてブラス(金管楽器)についてです。ブラスの発音方法はほかの種類の楽器とは異なっているのでDTMにはちょっとなじみにくいともいえますが，それぞれの楽器の音色の特徴や使われ方を考えてアレンジしてみましょう。

Taki Yasushi 瀧　康史

## §おすすめは「スペクトラム」

先月号で予告したとおり，今月はブラス(金管楽器)のアレンジについて考えてみます。

本論に入る前にまず，それに関係したCDを紹介しておきましょう。約10年ほど前に解散してしまったけれど，いまだにアルバムが出ているブラス系のバンド「スペクトラム」のアルバムです。

ご存じの人もいるでしょうが，このバンドの異色さは何といってもボーカルです。バックバンドは，いまでいうと米米クラブを思い起こすようなサウンド(まあ，ブラス中心ですし)なのですが，ボーカルはとにかく強烈です。初めて友人に紹介されて聴かせてもらったとき，「なかなか綺麗な声をしている女の人だな～」と思ったんですよ。で，友人とそのボーカルについて話をしていたら，「いい裏声しているでしょ？男なのに」って……。

そう。裏声がまるで女の人の声みたいに綺麗なんですよね。まあ，そういわれて耳を凝らして聴いてみると,「男の人の声だな～やっぱり」って感じるところはあるんだけど，これは一度聴いてみるだけの価値はあります。

まあ，それだけならば，ただのイロモノバンドですけど，そうではなく，ブラス中心のそのバックバンドのうまいこと。私は中学，高校とブラスバンド部には入らなかったし，それ以外の演奏経験もないのでブラスのことについてはあまり詳しくはわかりませんが，なかなかいい音を出していると思います。

それに，彼らは計2年しか活動しなかったせいかレパートリーは少ないらしいのですが，それでいてアルバムが6枚も出ています。そして，そのなかにはアレンジや，カバー，リミックスがいっぱいあるんです。これらはもとは別の曲だったりしたものを，うまくメドレーっぽくつないだりしていますから(最初にリミックスを聴いたときは，別の曲だということがわかりませんでした)，なかなか勉強になること請け合いです。

発売されているCDのなかでは，私は5枚目のアルバムが好きです。その筋の人に聞いたら，2枚目がかなり「おいしい」ということですが,今月は不況(私には関係ないような気もするけど)に続く財政難なので，残念ながらそちらは買うことができませんでした。

まあ，味があるのは確かなので，買ってみるなり，それ以外でも何かの機会があれば，ぜひ聴いてみてください。

## §ブラスの発音方法

「ブラスは楽器ではない」という人がいるのです。

私が「来月はブラスのことをやるよ」といったら，ラッパ吹き(とはいっても低音の楽器。私は彼のチューバが好きなのです)の友人がそう教えてくれました。

一瞬考え込んでしまったけれど，わけを聞くと，なるほどと思いました。

つまり，こういうことです。たとえばピアノなどの鍵盤楽器は内蔵されているハープのような弦が発音機関で，ヴァイオリンやギターは目に見える，あの張ってある弦が発音機関です。また，オーボエやクラリネットはいわゆる「笛」であり，息を吹き込むと音が鳴ってくれる機関(「リード」といい,それが震動して音が出るのです)があります。フルートやピッコロなどは，ほかの木管楽器とは原理がちょっと違っていて音を出すのにコツがいるので，素人がすぐに鳴らすのは難しいのですが，あれもやはり笛ですよね。

ところがブラスになると，楽器自体に発音機関はありません。ブラスバンドの人たちが大事そうに持っているさまざまな大きさの「マウスピース」(文字どおり，楽器の口に当てるところですね)は，それ自体にただ単に息を吹き込んでも音は鳴らないのです。

これは何をいっているかというと，発音機関を持つことイコール楽器であるというように定義するならば，ブラスは楽器ではない，ということになります。私は意味を漠然ととらえるほうなので，ブラスも十分楽器に見えますけれどね。

「発音機関がないってどういうこと？」そう思っている人がまだいると思うので，まずブラスについて，その発音原理も兼ねて説明しておきましょう(といってもたいした説明をするわけではありませんが)。

ブラスは形を見てもわかるとおり，入ってきた音をうまく共鳴させる形をとっています。つまりこれがすべてです。楽器そのものには発音機関はなく(笛の管の部分のように)，入ってきた音をうまく共鳴させて，それなりの音程に変化させる道具(楽器)ということになります。

発音機関はどこかというと，それは吹き手の口唇と声なのです。それを詳しく説明するとややこしくなるし，ここではそこまでの必要はありませんので，マウスピースのことは割愛します。しかし簡単にいうと，口唇が振動し，その音を拾って金属が共鳴し(もしくは管を変えて)音程を変え，あの甘い音や攻撃的な音などの，多彩な音が鳴るのだ，ということです。

口唇の振動は当然のことながら規則的ではないし，楽器そのものも蹴とばしたりすれば「メコッ」と凹んでしまう程度のやわらかい素材を使っているので，共鳴は必ずしも規則的ではありません。

いわばブラスは不規則に不規則をかけあわせたような条件のもとで音が出る楽器です。したがって，ブラスの音はコンピュータでは非常にシミュレートしにくいのです。そもそも，演奏者によって音が著しく変わる楽器ですし，たとえサンプラーだとして

も，人間が聴き比べてみて明らかに違うという音だけ取っても膨大な数になってしまいます。

科学が進歩して，ブラス楽器がコンピュータでシミュレートできるようになったとしても，それはブラス専用シンセであるとか，そういうオチが待っていそうな気もします。そんなものを使うよりも，うまく吹ける人を探したほうがいいような気もしますけれど。

ブラスの仕組みについての説明は，これくらいにしましょう。あまり突っ込んでしまうと，いえばいうほどボロが出てきそうですし(笑……としたいけど笑いごとではない)。簡単にまとめてしまえば，DTMで使うにはかなり難しい楽器ですから，それなりに覚悟して使うか，あるいは，初めからシンセブラスへと逃げるかするように，ということです。

最後にひとつ。「トロンボーン吹きがユーフォニームを吹くと，トロンボーンの音に聴こえる」とはどういうことかわかりますか？　私はなるほどと思ったのですが。

## §ブラスの種類

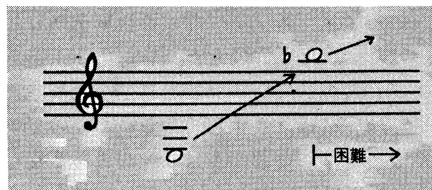
トランペットからはじまって，コルネット，フリューゲルホルン，ピッコロトランペット，フレンチホルン，トロンボーン，チューバ……と，ブラスの種類はうじゃうじゃあります。でも，残念なことに私が実際に吹いたことのあるのはひとつもありません。

というわけで言い訳がましいのですが，アンサンブルをとるのに楽で(私が楽なだけかもしれませんが)有名な4つの楽器，トランペット，ホルン，トロンボーン，チューバについてだけを簡単に説明しておきましょう。

**○Trumpet(トランペット)**

「ラッパ」といえばたいていの人がトランペットを思い出すとおり，最も目立ち，かつ最も吹き手がモテる(？)楽器です。そう，「スーパーマン」(死んじゃいましたね)のイントロや「スターウォーズ」のメインテーマの，おいしいところでメインメロディを担当し，ここ一発でかい音を鳴らして，聴衆の注目を集めてしまうあれです(いいなァ……)。

**図1　トランペットの実用音域**




音域はO3E～O5B♭で，これ以上は絶対に不可能ではありませんが困難になります(図1)。そうそう，記譜はB♭です。これは下加線第一がO4Cではなく，O3B♭ということです。金管楽器はたいてい，このように記譜と実音が違うので，ダマされてしまうと，「なんだこりゃ」ということになりますから，ご注意。

**○Horn(ホルン)**

ホルンというのは，「角」のことを指すのであって，決して「巻き貝」のことではありません。いや，笑いごとではなくて，私も「Horn」というのは，あのフレンチホルンのインパクトがあまりに大きかったから，巻き貝のことをいっているのかとひそかに思っていた時期もあったのです。

巻き貝に似た形をしているので，オーケストラを見回せばすぐ発見できると思われるホルンですけれども，この楽器は金管楽器にしては，ちょっと異色の音色をしています。

ホルンだけの音を聴いてみるとわかりますが，この楽器は正弦波に近い，ブラスとしてはかなり整った音をしています。悪くいうとやや味気ないのですが，よくいうと「上品」といったところです。ブラス特有のケバケバしさがなく，上品な，何というか含んだような音がして，非常に滑らかに聴こえる音色の楽器なのです。ともすれば分散してしまいがちなブラスのアンサンブルを，中音域で広がりのある音できゅっとまとめてしまうことができるような感じがします。

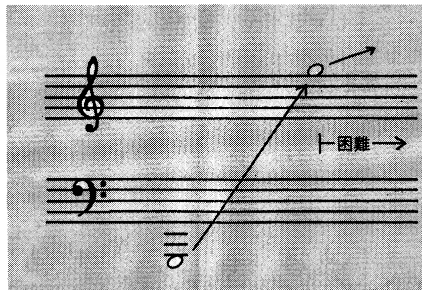
ホルンは，ラベルの「古風なメヌエット」やエルガーの「威風堂々」などでメロディを奏でています。それらを聴くだけで，この楽器の存在価値がわかりますよね。

音域はO2F～O5C(図2)。これ以上は困難になります。記譜はO2E。楽譜上はO3CがO2Fになるので注意すること。

**○Trombone(トロンボーン)**

ボレロにはありそうな気がしますが，トロンボーンそのものはメインメロディを奏でることはあまりないように思います。

**図2　ホルンの実用音域**




まあ，中低音域の楽器というせいもあるんですが，あまり目立たなくて，たとえば曲に厚みを持たせるとか，オブリガードを吹くとか，バスの楽器とはまた違った意味での縁の下の力持ち的な存在ともいえるでしょう。

一般には1st, 2nd, 3rdと，3本(3種類でいっぱいのときもあり)中低音というか低音あたりで，かなりパワーのあるフレーズを鳴らしています。まあ，誌面上では「この音だ！」というふうにお聴かせしてみることはできませんが，スターウォーズのメインテーマなどを聴いていると，低音部で「ばばばば」と気合の入ったフレーズを吹いているので，注意して聴いてみればわかることでしょう。きっとわかると思うなぁ……(逃げモード)。

管がスライドするようになっていて，このスライドに7つのポジションが決められ，それらが半音ずつの音程差がある。とまあ，そんな仕組みの楽器ですから，当然，グリッサンドは最大増4度(金管楽器にしてはかなり広いほうです)になります。ほら，ガイーヌの「剣の舞」の「ぱぁ～あぱぁ～あ」ってやつですよ。え？　わからん？　いや，剣の舞を口ずさんでいけば，その箇所にきたら，「ああ，これか」とわかりますよ，きっと。

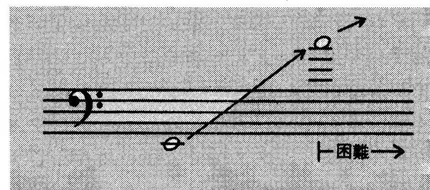
楽器そのものを見たことがある方は，すぐに理解してくれると思いますが，このスライドのポジションが遠くなれば遠くなるほど，連続して発音しにくくなります。まあ，物理的に困難になってしまうわけですが，コンピュータで音を作る場合にこれらを厳密にとらえていくときりがありませんし，それ以前に私の知識も不明瞭だったりするので，このへんは勘弁してください(たいていはメロディを作ったあとで，これできる？　って誰かに聞いてしまう場合が多いのです。実は)。

音域はO2E～O5C(図3)で，これ以上は困難です。

**○Tuba(チューバ)**

はっきりいって，私はブラスのなかではこの楽器が一番好きです。小学校のときは

**図3　トロンボーンの実用音域**



ひそかに,「中学校に入ったらあのおっきなチューバを吹くんだ！」と志していたのですが，結局でかい身体のせいでバスケット部に勧誘されてしまいました(そんなことどうでもよいが……)。

音も低くて，ストリングスにたとえればコンバスのような存在ですから，当然メロディアスなフレーズはあまりありません。ほとんどベースと考えてよいでしょう。トロンボーンとうまくミックスして使われる場合が結構あります。

あまり一般的ではないと思うのですが，チューバにソロをやらせるとすごくエッチだよということだけ，声を小さくしてお話ししておきましょう。

音域ですが，チューバは基音となる音が管によりそれぞれ音域が違います。ここでは一番ポピュラーなB♭チューバの音域だけお話ししておきましょう(というか，私はそれ以外は知らないのです。基音がわかるから予想はつきますけれど)。

B♭チューバの音域はO1B♭～O4B♭までです(図4)。

## §ブラスの使われ方

もちろんどの楽器でも同じことですが，作曲者の好みや，大まかにいって曲のジャンルによっても，ブラスの使われ方は大きく違ってきます。

マーチでの使われ方とシンフォニーでの使われ方，ジャズでの使われ方，ポップスでの使われ方，フュージョンでの使われ方などは，まったくといっていいほど違いますよね(このうち比較的，ジャズとポップス，フュージョンは互いに境があまりないくらい似かよっていますが)。

今回は，このなかで最も簡単（と私は信じている）シンフォニーでの使われ方を主にお話ししていきましょう。

ジャズでの使われ方，フュージョンでの使われ方については，説明したいのはやまやまですが，それらについての知識にはまだ欠陥がかなりあることから，今回はパスさせてください。資料と知識がたまったら，いつかやってみたいと思います。

さて，シンフォニーでのブラスの使われ方は，1992年8月号のこの連載で紹介したCDシンフォニー版「ソーサリアン」をよく聴いているとわかります。シンフォニーのメロディパートはたいてい，木管楽器，弦楽器が担当する場合がほとんどです。よくよく聴くと，ブラスなどはここ一発の盛り上がりでないかぎり，ほとんど登場していないようにも聴こえますが，実は合いの手(このいい方は友人T氏に教えてもらったのですが，実に的を射ていると思います)で入っています。そう，演歌などの合いの手とまったく同じです。メロディが長いところで，すかさず「ばーばーばー」と入るといった感じです。そういうことに気をつけてよく聴くと，たいていのシンフォニーではその程度の使い方しかしていないことがわかります。

しかし，ブラスはさっきもいったとおり，ストリングスなどに比べると「ここ一発！」のインパクトがはるかにあるので，盛り上がりなどには欠かせません。ドヴォルザークの「新世界から」などは，ほとんど，聴き手を驚かすために使ってるんじゃないかと思われるほどの気合の入った使われ方をしています。

一方，ラフマニノフとか，あのあたりの人は，「ブラスなんか鳴ってるの？」と思ってしまうくらい，ブラス楽器を使っていません。いや，ホントはところどころで使ってはいるのですが，あまりにも前面に出てこないので，まったく気づかれないことも多いというわけです。まあ，使い方とすれば，縁の下の力持ち的なコードの補佐や，合いの手……しかし，これもあんまりないのですよね。

このブラスの使い方を聴くだけで，作曲家の趣味がわかるので，その点に注目していろいろな曲を聴いてみるのも面白いでしょう。

## §ブラスでのハーモニー

ブラスでハーモニーを奏でる場合，音の特色上，コードの配置を図5に示すような密集配置にしなくてはなりません。これは別名クローズボイシングともいうのですが，ブラスという楽器がなぜ開離配置にしてはならないかは，実際にやってみればわかります。

ブラスの音は，不安定な発音によってできているにもかかわらず，意外と音色は正弦，というか鋸波に近くて，非倍音をあまり含んでいません。そのため，コード上にすきまを与えたときに，ストリングス系の楽器ならば音色の厚みがそれをうまく埋めてくれるのですが(図6)，ブラスではハー

図4　チューバの実用音域

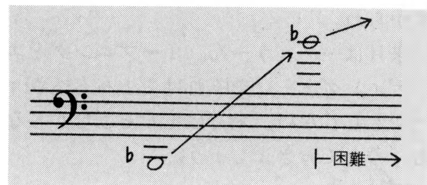
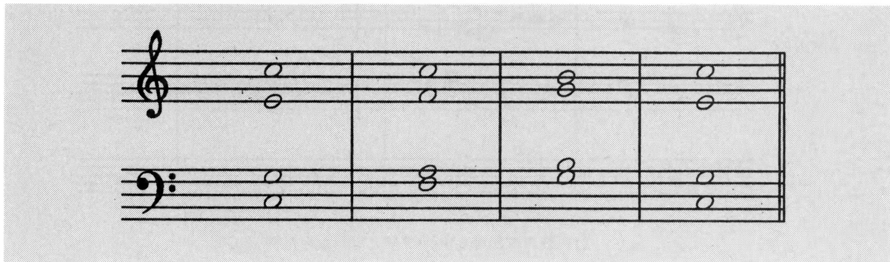



図5　ブラスの密集配置
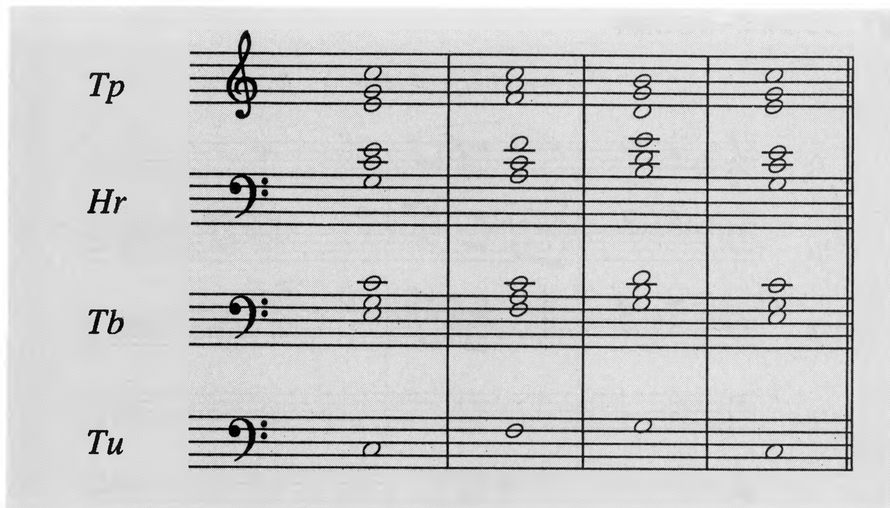



図6　ストリングなどの開離配置の例



モニーに欠陥ができてしまいがちになるのです。

そのため，ブラス，特にトランペットやトロンボーンなどでは，３度や５度でのハモりをやる場合がほとんどになります。

ここで注意してほしいのは，図５に示した形は四声体ではないことです。というよりも，１つひとつの楽器ですでに３つの楽器を使ってしまい，それひとつでコードの構成音をすべて鳴らしてしまっているのです。

このくらい重厚にやらないと，ブラスのハーモニーはかなり不安定になってしまいがちです。

ただし例外はホルンで，ブラスなのにもかかわらず音に広がりがあり，お互いの音がハモりやすいので，うまく中音域に当てはめれば，それなりのハーモニーを醸し出してくれます。まあ，これらは図５を見ながら考えてみてください。

## § 実際にはどのように使われているか

図７は，私が昔作ったシンフォニーから抜粋したものです。もともとは22段譜なのですが，それではとても見にくいし，ここでの説明にはすべてのパートは必要がないので，ここにはその中からブラスとストリングスの部分だけを抜き出してみました。

図7　シンフォニーからの抜粋

Main Theme for Almine

Tp

Hr

Tb

Tu

Strings

22段譜のなかからBrass 4 段とStringsを抜粋

ストリングスは ヴァイオリンからコンバスまで全部で４段で，図７の下のほうの４段にそれぞれ示してあります。

ハーモニーの構成は，ブラスとストリングスがあるので，対比してチェックできると思います。

これはサビの部分なので，メインメロディはトランペットとストリングス，本来ならばここへフルートとクラリネットなどがユニゾンしますが，それについては割愛します。同じメロディが何度も繰り返されて，そのたびにアレンジが変化するというのがシンフォニーの醍醐味ですが。

１小節目には，音符にテヌート（図中①）が付いています。ふつう，テヌートというのは，スラー（②）とほぼ同じですが，ブラスの場合は，テヌートでは音をそれでも１音１音切るのに対して，スラーでは一息で続けてしまう……もっとくだけてしまえば，テヌートは１音１音に弱めのアタックが残るけれど，スラーはアタックがほとんどないともいえます。

ホルン，ヴィオラは完全にオブリガード，それから３度，５度下のハモりにまわっています。メロディが白玉（２分音符より長い音）か，あるいはそれに近い長さのときに裏に合いの手のように入れてあるというところです。

トロンボーンは厚みを持たせるためのパートです。なにせ，最大の盛り上がりのところですからね。１小節目はバックティンパニがバラバラと入っているくらいですから。③の部分の斜め線３本は半音下からのポルタメントを表します。

DTMで打ち込むならば，極端なほどこの音の長さを意識するのが鉄則だと思います。スタカートがかかるなら，ゲートタイムをうんと短く，テヌートならば長く。そうすれば，それなりにブラスらしく聴こえるのではないでしょうか？

チューバ，コンバスはベースノートをやってるだけですから，あえて説明は必要ではないですよね。

が，しかし……。ひそかにチェロが怪しいことをやっていたりするのですが……。わかりましたか？

## § おわりに

本当は，詳しい友人に頼んで，自分自身でもブラス楽器を吹けるようになりたいと思っていました。もちろん完全にマスターしようとは思ってはいませんが，多少なりとも自分ができるとできないとでは，かなり違ってくると思ったからです。実際に楽器を演奏してみて勉強しようと思ったのですが，時間の都合でその計画も無残に崩れ去ってしまいました。

しかし，ブラスについてはまだやり残したことがいっぱいあります。ジャンル別の使い方についてなどもやってみたいので，まだまだ私には研究の余地があります。とりあえず，知っているといないとでは結構違うよ，といったようなところをまとめてみましたが，それなりに役に立つならば幸いです。

いつになるかわかりませんけれど，この続きはかならずやりますので首を長くして待っていてくださいね。

さて。

私はいま，友人からもらったエレキギターをちょくちょくやっています（仕事をサボりつつ……）。これは，ギターの曲が書きたいからなのですが，いろいろまとまったら，また誌面でその情報を流したいと思います。

なんでも，実際にやってみるのが一番ですからね。

来月は……。う～ん。オープニングとエンディング？　いやそれはちょっとヘヴィーかもしれない，来月は忙しそうだからなあ。考えておきましょう。

ではまた。

特集

# D.I.Y.ハードウェア

ソフトウェアとハードウェアは表裏一体，とはいっても，処理速度などの性能面ではハードウェアのほうが分がいい。その代わり，誤動作したときのリスクはソフトウェアよりはるかに大きい。
ハードウェア拡張の持つ可能性は非常に大きい。さまざまなものに接続してパソコンの用途を広げたり，いろいろなものをコントロールして遊んでみたり，そしてコンピュータ自体の機能を拡張したり，性能を強化することさえできる。
しかし，ソフトウェアでさえシステムに密着した部分は手を触れないほうがよい，とされる。ましてやシステムに近い部分のハードウェアなどは一般の人の手に触れられるべきではない。
すでにソフトウェアについてはかなり突っ込んだところまでユーザーの手にかかり拡張されつつある。システム拡張のためのソフトウェアも発売されている。しかし，システムを拡張するハードウェアについてはほとんど例がない。これも，ある意味で異常なことなのではある。
できるだけ安全なかたちでのハードウェア拡張の指針を探る。そういったことも必要なのかもしれない。

CONTENTS



Hardware

アクセラレータを作る(その1)

# 68020ボードの構想

Ishigami Tatsuya 石上 達也

68000ファミリーは基本的にはオブジェクトコンパチ。微妙な違いはあるのですが，ほとんどのプログラムがそのまま使えます。ここではX68000用に68020アクセラレータを製作してみます。

「Creative Computer Music入門」の瀧氏との会話。

「石上さん，最近のDOS/Vマシンて競馬みたいだと思いません？」

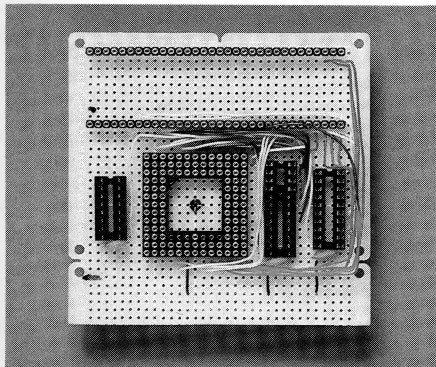
「え？」

とっさには理解できなかったのですが，「おっぉー，第3コーナー手前，先頭はインテル，すぐその後にサイリクス，AMDチップス&テクノロジー。サイリクス迫る，サイリクス迫る。おっと，ここで，インテル急加速だー」

と，ただひたすら速度を競っている競争馬のような状態を指しているようです。これは，けだし名言かもしれません。うむ。

それから，最初に断っておきますが，X68000+SX-WINDOWというシステムはそれほど遅くはありません。DOS/Vマシンはいくらクロックを上げても，ソフトウェアのほうで，それ以上の速度を要求してくるから，競馬みたいなことが起こるわけで，X68000にはそのような必要はありません。私は486DX-33MHzにS3カードという聞く人が聞くと涎を垂らしそうなマシンも使っているのですが，ウィンドウを開いたり閉じたり，ファイルをコピーしたり，メニューを選んだりという通常の動作では10MHzのX68000ACE上のSX-WINDOWのほうが遥かに快適です。

そんなわけで，実はこのアクセラレータというのはそれほど必要な回路ではありません。では，なぜ必要もないのにそんなものを作るかというと，それはそれ「パーソナルコンピューティング」です。


今回は基本配線のみ

## アクセラレータを作る

アクセルというのはブレーキの反対で速度を上げるものです。これは実はAcceleratorの略で，辞書を引くと「加速装置」と出ています。一般になにかの速度を上げるものがアクセラレータというわけです。ここではコンピュータの計算速度を上げるものを意味しています。

車の場合はエンジンの回転数とギヤ比で速度が決まりますが，X68000にもそれに相当する要素があります。エンジンの回転数に相当するのがクロック周波数で，ギヤ比に相当するのが，1命令あたりのマシンサイクルです。今回は，そのうち後者のほうをちょっとばかりいじくってみます。また，余力があったら，前者のほうにも手を出してみることにします。1命令あたりのマシンサイクルを上げると，いままで，16ビットの割り算を行うのに最大158サイクル要していたのが，25サイクルになるなどの効果が表れてきます。

さて，このような装置を作成するのは，大仕事です。さすがに，1回の製作記事ではフォローしきれません。また，正直にいって，モノが完成して自宅でビュンビュン動作しているわけでもありません。まだモノは私の頭の中にしかないのです。どちらかというと，この連載は失敗するかもしれないし，成功するかもしれない回路製作の中間発表的なスタイルで進めていきます。ひょっとしたら，私の考えているよりも，もっとうまい逃げ道があるかもしれません。皆さんのなかで，よいアイディアをお持ちの方がいらっしゃいましたら，ぜひとも編集部までご一報ください。

## 68020を使う

カタログ類をパラパラとめくると，どうやら68HC000に16MHzバージョンよりも速いものはないようです。となると，頭をもたげてくるのが68000ファミリーのより上位機種ということになります。

まず，68000にいちばん近いものということで，68010。これは，1983年に作られた，どちらかというと68000のバグフィクスバージョンという意味合いが強いLSIです。バグフィクスといっても（真の意味での）マルチタスクを行わせようとしたり，バスエラーなどからソフトウェア的に回復しようとしないかぎり，68000にバグは見つからないのでご安心を。

また，仮想記憶をサポートしていたり，内部ロジックの最適化などにより，68000と同じ動作を行わせても，数%～数十%速くなったりします。が，X68000を改造の危機にさらしてまで高速化させる価値のあるほどには上がらないので，パス。

ただし，ピン配置がほとんど68000と同一ですのでクロックの高速化と同程度には面白い実験かもしれません。X68000がC-MOSで組まれているのに対し，68010はN-MOS版しか用意されていないようです。ある程度のリスクは覚悟しなければならないでしょう。

68000ファミリーのなかでは最上位に位置するのが68040です。Macintoshでも Quadraシリーズに採用され，その怪物パワーはClassicシリーズなどを寄せつけません。しかし，バス方式が非同期式から同期式に変わったり，電気的特性が厳しかったり，タイミング設計が非常にクリティカルだったりするので，これもパス。だいたい，このLSIはアマチュアの手の出せる代物ではありません。

68040がだめなら，2番手の68030を考え

てみます。この68030が発表されたのは1987年ですから，もう５年も前のことになります。さて，特徴として，

1)　バーチャルメモリをサポート
2)　32ビットデータバス
3)　32ビットアドレスバス
4)　256バイトの命令キャッシュ
5)　実効アドレスモードのサポート
6)　ビットフィールド関係の命令の追加
7)　256バイトのデータキャッシュ
8)　MMU(Memory Manage Unit)の内蔵
9)　高速バスサイクルのサポート

などが挙げられます。ちなみに，1)が68010，6)までが68020で新たにサポートされた機能となっています。残りが68030ならではの特徴です。

バーチャルメモリというのは，実際には少ない実装メモリをハードディスクの内容ととっかえひっかえして，アドレス空間いっぱいに(この場合は$2^{32}$＝4Gバイト）実装されているように振る舞う機能のことです。が，この機能を使うには，ハードウェアをそのように設計しなければいけないし，OSも作り直さなければなりません。当然，私にそんな実力はないので，この機能は見送ることにします。ただし，いずれ手を打つかもしれないよ，と伏線を張っておきましょう。

2)，3)の32ビットデータ&アドレスバスというのは，68020&68030が正真正銘の32ビットMPUであることの証明にほかなりません。68000も内部的には一応は32ビットMPUですが，この内部で作られた32ビットのデータを外部に出力するときに16ビット×２回に分割してしまうので変則的な存在です。68000用に設計されたX68000も，16ビットデータバスしかサポートしていないので，これに68020や68030を搭載しても結局は16ビット×２回に分けて取り扱います。結局，この特性は今回の製作においてまったく生かされないことになります。ただし，メモリぐらいはアクセラレータ上に32ビットデータバスを張って，専用に設けてもよいかな，と思う今日この頃です。

4)の命令キャッシュというのは，いちいちメモリに命令を読みにいかなくても，MPUが一度読み込んだことのある命令は記憶しておいて，その記憶を利用しようという機能です。１回読み込んだ命令はしばらく覚えておいて，後々その内容が必要になったときに備えておけば，メモリのアクセスを１回減らすことができますね。１回メモリアクセスを減らせば，その分だけスピードの低下を防げるという機能です。その覚えておける容量が256バイトですよ，というのが4)の意味するところです。命令だけでなく，メモリ上のデータについてもそのようなからくりを採用しようというのが7)のデータキャッシュです。

5)，6)は，プログラミング上，そういう命令も使えるようになりました，ということです。X68000を改造して既存のプログラムを走らせる分にはまったく関係ありません。ただしgccの一部のバージョンでは，これらの命令をサポートすることもできるため，それなりの高速化が望めそうです。

8)のMMUというのは，UNIXなどのマルチタスクOSをサポートするには必要なのですが，とりあえずHuman上で使っている分には関係はありません。よって，パス。ただし，仮想記憶とともに，伏線を張っておきましょう。

さて，68000ファミリーは従来非同期式と呼ばれるインタフェイス方式を採用してきました。MPUが「いま，アドレスバスにデータを載っけたよー」と合図をして，まもなく「データバスにデータを載っけたよー」と合図をします。これを見たメモリが，やおら腰を上げそのデータを読み込みます。読み込み作業が完了したら，「動作を終了したよー」と合図をMPUに戻すのです。なにを行うにも動作主体からの合図がないと次のステップへは進めないインタフェイス方式です。

ちょっと考えればわかるように，各部品は同じスピードで動作している必要性はありません。今回の製作やX68000XVIでのクロックの高速化もこの性質を利用しています。しかし，クロック周波数が20MHzあたりを境に，この合図を送る時間も馬鹿にできなくなってきます。特に，メモリからMPUへの応答信号はオープンコレクタ

## キャッシュをめぐる環境

キャッシュシステム。すでにどこかで聞いたことのある単語ではないでしょうか。そう，アクセス速度の遅いデバイスに対し，代わりに速いデバイスをその一部にあてがってやるアレです。MPUまわりのキャッシュ機能はX68000にはありませんが，半導体メモリよりももう一段遅いデバイス（フロッピーディスクとかハードディスクとか）に対し，それらよりは速い半導体メモリをあてがってやるような話はどこかで聞いたことがあるでしょう。

残念ながら，今回のアクセラレータで使用する68020にはプログラムキャッシュのみで，データキャッシュがありませんが，そこらへんの話題をちょっとばかり見ていきましょう。

プログラムキャッシュとは，MPUにプログラム（のコード）を一度読み込ませれば，それを忘れるまでメモリに読みにいかないということです。つまり，短い範囲でのループなどはひと回りすると，２回目以降はプログラムをメモリに読みにいかずにすむので，圧倒的に処理速度を上げることができるようになるわけです。

で，この短い範囲のループのもっともよく使われる場面というのは，なんといっても「データの転送」なわけです。たとえば，

```
loop:
    move.b  (a0)+,(a1)+
    dbra    d1,loop
```

などというプログラムをプログラムキャッシュなしで実行した場合，

1)　まず，「move.b　(a0)+,(a1)+」（のプログラムコード）をメモリから読み込む
2)　アドレス(a0)からデータを読んできて，それをアドレス(a1)に代入する
3)　「dbra d1,loop」（のプログラムコード）を読み込む
4)　1)へ飛ぶ

の繰り返しとなるのですが，プログラムキャッシュを作動させた場合，２回目以降のループは，

1)　アドレス(a0)からデータを読んできて，それをアドレス(a1)に代入する
2)　1)へ飛ぶ（外部メモリへのアクセスはなし）

だけですから，これはもう速くなるわけです。

そして，データの転送というと思い出されるのが，DMA（Direct Memory Access）です。DMAコントローラというのは，平たくいってしまえばデータ転送を高速に行うためにMPUを特化させたものですから，MPUにこんなにデータ転送を高速に行われては立つ瀬がないのです。

さらに，MPU自身にデータキャッシュ機能があったりすると，DMAコントローラがそれに内緒でメモリ内容を書き換えると，キャッシュ内容を実際のメモリの内容と同一に保てなくなってしまいますので（コヒーレントが保てなくなるという），いろいろ辻褄あわせを行ったり，ひどいときにはDMA転送を行うたびに，データキャッシュの内容をきれいさっぱり忘れさせてしまうのです。

このように，キャッシュ機能は非常に強力なのですが，DMAなどの周辺デバイスとのバランスをうまくとって設計してやらなければ，それを生かすことはできません。さっきから，なんのことをいっているかというと，実はIBM-PC/AT互換機の悪口をいっているのですが，この機械はさらに具合の悪いことに，外部デバイスをアクセスするときに，大昔の機械との互換性を保つためクロック周波数が8MHzまでグググッと下がってしまいます。つまり，DMAを使うとMPUでデータの転送を行うときよりもかえって時間がかかってしまうのです。

そんなわけで，巷ではひたすら互換機の速度を測っているようですが，実はクロック周波数を測っているだけだったりするのです（ただし，最近の本家IBM機やEISAバス機などは，ちゃんとここらへんが設計されているので安心の模様です）。

それからIOCS.Xなんかをのぞくと，同じような命令がずらーっと並んでいて，「あ，これはループを展開したな」などと思うのですが，プログラムキャッシュを搭載したMPUの場合には，ループは展開しないほうがお得なのはわかりますよね。ま，ここらへんの話は試作機１号が動作してからむし返すことにしましょう。

とかオープンドレインと呼ばれる，非常に立ち上がりが遅いという電気的特性を持った素子で構成されている信号線なのです。

で，68030には，このようなジレンマを解消すべく同期式インタフェイスが変則的に用意されています。全体の動作を８分の８拍子よろしく制御するのです。１拍目には，アドレスを確定して，２拍目には，データを確定して……のように制御すれば，合図の遅れは気にならなくなります。これが同期式インタフェイスです。68000にも同期式インタフェイスをサポートする信号線としてＥクロック端子が用意されています．が，これはまた別物です。詳細は後述しますが，「歴史は繰り返す」の見本みたいな代物です。

例によって，この同期式インタフェイスをサポートするにはいくらMPU がひとりで頑張っても駄目で，メモリなどがこれに合わせて設計されている必要があるのです。よって，この機能も今回は見送ります。

さて，68030の特徴は以上ですが，もうひとつ落とし穴がありました。68030ではキャッシュを使用できるのは，外部バスが32ビットのときのみなのです。68020に比べて，数割の高速化が図られているとはいえ，キャッシュほどに威力があるとは思えません。非常に残念ですが，「今回は」MPUに68020を使用することにします。キャッシュの威力については囲みを参照してください。ただし，Macintosh ClassicIIでは68030を16ビットバスで使用しているのに，かなりの高速化が図られているようですが，どのようなカラクリがあるのか私には見当もつきませんので，やっぱり68020にします。

ここで，「ほほう，アクセラレータに68020を使うのか。おい，ちょっと待てよ。確か68020には組み込み制御用に68EC020というのもあったよな。あれなら，実売価格はともかくとして，一応代理店価格は68020よりも安いっていうしな。組み込み用に省略されたっていったって，アドレスバスが24本になっただけだしな。確か，パッケージも97ピンでちょうどいいサイズだったな」と考えたあなたはスルドイ。どのくらい鋭いかというと私の代わりにアクセラレータを製作してほしいほど鋭い。

で，なぜ68EC020を使用しないかというと，バスアービトレーションの制御が68020の$\overline{BR}$，$\overline{BG}$，$\overline{BGACK}$の３本方式（3-wired-arbitrationという）から，$\overline{BR}$，$\overline{BG}$の２本方式（2-wired-arbitration）に変更されたためです。68EC020のユーザーズマニュアルにも，ここらへんのインタフェイスについて記述されているのですが，まぁ，最初は問題が少なそうなところから始めようということで，68EC020の採用は保留状態ということにさせていただきます。

今回の製作が順調に進み，ソフトの制御もうまくいったら（そう，実はハードウェアだけでなくソフトウェアのほうもいろいろいじってやらなければならないのだ），そのあとにクロックの高速化とともに，考え直すことにしましょう。

## ダイナミックバスサイジング

前述のように68020は32ビットMPUです。しかし，X68000の内部には16ビット幅のデータバスしか走っていません。これをどうやって接続するのでしょうか。その解答が，ダイナミックバスサイジング機能です。アメリカ製日本語に訳すると，「ダイナミックにバスのサイジング（SIZE＋ing）を行う」というのですから，接続するバス幅は，どんどん変更してよいのです。８ビットだろうが，16ビットだろうが32ビットだろうがかまいません。

当然，数が大きいほうが一度に転送できるデータも多いので，そちらのモードのほうが動作スピードは速くなります。よって，できる限りシステムは32ビットバスで組んでおきたいところです。しかし，プリンタやシリアルポートのように８ビットバスを基本単位として動作するデバイスも多くあります。ROMを焼くときも，できれば８ビットで済ませたいなどと考えることもあります。

それらを68000で実現しようとすると，$\overline{LDS}$や$\overline{UDS}$をシコシコやって，非常にメンドくさかったのですが，このダイナミックバスサイジング機能を用いることによって，信じられないくらい簡単に行えるようになります。ついでといってはなんですが，68000で禁止されていた「ワード境界をまたぐメモリアクセス」が，68020では解禁となっているのは，半分くらいこの機能のおかげです。

せっかくの機能ですが，X68000本体はバス幅＝16ビット固定で組まれているので，当面はこの機能の一部しか用いないで16ビット固定でやっていくことにします。

具体的に，68000を16ビット幅固定で使用するには，68000でいうところの$\overline{DTACK}$を$\overline{DSACK1}$に，従来のデータバスを68020のD16〜D31に接続し，$\overline{DSACK0}$を＋5Vへプルアップします（しかし，回路図ではそのようになっていません。これは，あることのお守りなのですが，それは来月号で説明します）。

## その他の変更

バス幅の調整法はわかりました。では，あとなにをすればよいのでしょうか。信号線の多くは同じファミリーだけあって，直結でいいのですが，なかには機能の増えたり減ったりした信号線や，削除されたり，追加されたものもあるので，面倒を見てやらねばなりません。

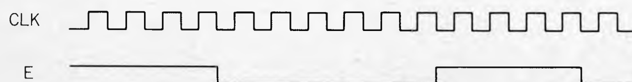
**●Ｅクロック**

68000ファミリーでも最近のものになると，同期式インタフェイスを採用すると話しましたが，この方式は決して新しいものではありません。Z80がまさにその証拠ですが（さらには8080も），モトローラの前作6800シリーズも，実は（というほどのものでもないが）同期式インタフェイスなのです。さて，68000発表当時は，非同期式インタフェイスをサポートする周辺LSIはほとんどありませんでした。市場に出回っているパラレルポートやシリアルポート用のLSIは，みな同期式インタフェイスで接続しなければなりませんでした。そこで，暫定的な処置として68000も非同期式インタフェイスをサポートするための機能が付け加えられたのです。そのひとつがこのＥクロックです。中身のほうはクロック入力をデューティ比６：４で10分周し，出力するというものです（図１参照）。

うまいことに68000の$\overline{BGACK}$をアサートし，機能を停止させても，この出力はカットされないので，68000をＥクロック発生機として用い，この出力を使用することにします。

X68000の内部をのぞいて見るに，同期式インタフェイスの必要そうなLSI（つまりは，6800ファミリーのLSI）は見当たらないので，省略してもよさそうなのですが，ひょっとしたら，拡張ボードで使ってたり，カスタムチップの中で使っていたりするかもしれません。縁起ものとして，一応はサ

図１　Ｅクロック



ポートしておくことにします。

そんなわけで，アクセラレータ上には，いらなくなった68000を載せておくことにします。うまくいけば，スイッチひとつで68000モードと68020モードの切り替えを行えるようにできるかもしれません。

**●$\overline{\mathrm{VPA}}$，$\overline{\mathrm{VMA}}$**

前述のEクロックと同様に，同期式インタフェイスをサポートする端子として，$\overline{\mathrm{VPA}}$，$\overline{\mathrm{VMA}}$端子があります。これは6800ファミリーのマッピングされたアドレスを68000がアクセスすると，アドレスデコーダは$\overline{\mathrm{VPA}}$をアサートし，これからの動作が6800型の同期式インタフェイスであることをMPUに知らせます。そして，その了解の印に$\overline{\mathrm{VMA}}$をMPUはアサートします。

また，$\overline{\mathrm{VPA}}$には，この機能とは別にオートベクタ割り込みの要求受け付けという機能があって，両者を区別するには，FC0～FC2端子とアドレスバスA16～A19に注目しなければなりません。前者の端子がすべてHになっていれば，MPUは割り込み応答サイクルなので，$\overline{\mathrm{VPA}}$はオートベクタ割り込みとして，そうでないときには同期式インタフェイスの転送要求用として使われていることとみなせるので，これを利用します。

## GALについて

図1～3などでは，回路図中にハードウェア入門などでお馴染みの記号が使われていましたが，これらをまとめたはずの図4では，なぜかこれらの記号は姿を消してしまっています。その代わりに，なにやら見慣れない16V8というLSIが見受けられます。

さて，この16V8とか20V8などというのが，巷で流行のGALという奴です。GALといっても，

「あ，ごめーん，オレ今日これで帰る」

「はは～ん，さてはGALだな」

でなくて，General Array Logicの略です。言葉だけではなんのことか，よくわからないと思いますが，とにかく，図を見てください。

「なんじゃ，こりゃー。こんなのほとんどの信号がショートしちゃって，使い物にならないじゃないかー」

と思ったあなたは正しい。そのとおり，このままではまったく使いものになりません。そこで，左半分のスダレのような部分を（電気的に）ブチブチと切って，目的のロジック回路と同じものにするのです。つまりGeneral（汎用的な）Array（配列）のロジック回路なのです。

これだけなら，PLD（Programable Logic Array）と呼ばれる素子ならば，たいていは実現できるのですが，さらにGAL は偉いことに，一度切ってしまった部分を修復できるのです。これはスゴイ。というわけで，今回のように，とにかく動かしてみないとなにが起こるかわからない，しかし，回路を小型化しなければならないし，実装密度もかなり高いので，あまりワイヤを取ったり付けたりを繰り返すこともできない……というような用途には打ってつけです。さっそく，採用しました。

で，このGALという素子をお店から買ってくること自体に問題はないのですが，そのあとで今回のアクセラレータ用にいらない部分をブチブチ切り放してやらなければなりません。これには高価な装置が必要となってきます。ちょうど新品のROMを買ってきてもROMライタがないと使えないのと似ています。

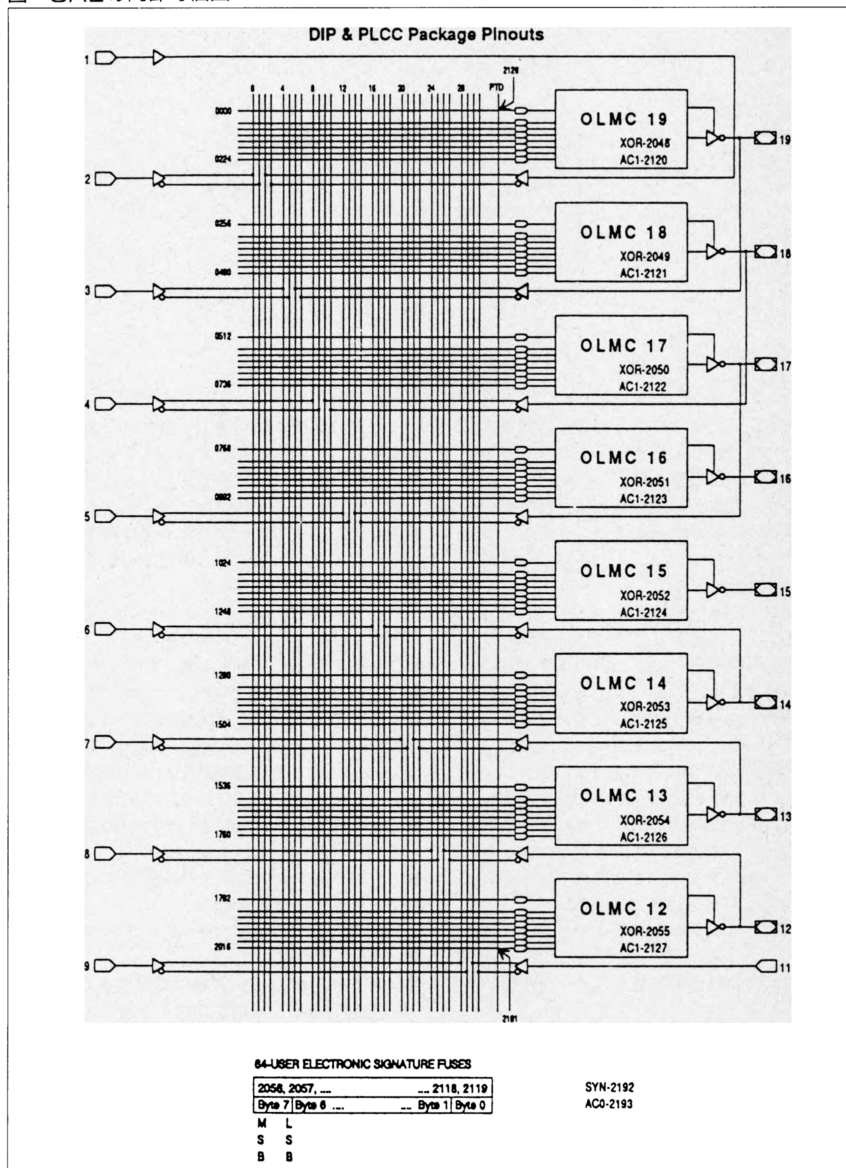
参考までに，私はこの装置（GALライタという）に，K.E.M Electronics社製のKEM-907GAL（「やきいもGAL」）を使用しましたが，価格が約5万円かかりました（ただし，編集部持ち）。

また，投資額はこれだけでいいかというと，世の中厳しいものでまだまだあります。「ココの端子とココの端子はこのような関係にしたい」というデータを，「そのためには，ココとココとココに電圧をかけて，配線を焼き切ってしまえばいいよー」というようなデータ（ヒューズパターンという）を作成するコンパイラが必要となってきます。私は，このコンパイラに同じくK.E.M Electronics社製のMini-Cuplを用いました。値段のほうは32,000円です（ただしIBM PC用）。

おそらく，この組み合わせが日本で入手できるGAL開発セットのなかでいちばん安価なものです。確かに，値段以上の働きをする機能を持ってはいるのですが，読者の皆さんはアクセラレータの製作1回のために，8万円近くも投資するわけにはいかないでしょう。

そんなわけで，この短期連載が成功裏に終わった暁には，親切な石上が，ちゃんと読者の皆様に実費で書き込み済みのGALを配布する予定ですので，ひとまずご安心ください（でも，DōGAの連載読むと，配布手続きってメンドくさそうだなぁー。ま，いっか）。

図　GALの内部等価図



### ●$\overline{AVEC}$

68020には前述の$\overline{VPA}$端子がなくなった代わりに，この$\overline{AVEC}$端子が用意されました。この端子がアサートされれば，MPUはオートベクタ割り込みとみなすので，$\overline{VPA}$がオートベクタ割り込みとして使用されたときにはこの端子をアサートすることにします。

以上をまとめると図2のような回路になります（待てよ。$\overline{AVEC}$は$\overline{VPA}$直結でもいいのかな？　答えは来々月号くらいに)。

### ●$\overline{RMC}$

68000シリーズには，リードモディファイサイクルというのがあります。なんのためにあるのかを説明しだすとキリがないので，ここではやめておきます（ヒントはマルチタスクのセマフォ)。

さて，このサイクルではバスをロックした状態で，1サイクル内にメモリからの読み込みと書き込みをいっぺんに行います。68000では，このとき$\overline{AS}$をアサートにしっぱなしで，この状態にあることをほかのデバイスに示したのですが，68020ではこのサイクルでバスをロックするのに$\overline{RMC}$という専用の出力端子が追加されました。もちろん，専用の信号が用意されたのですから，このサイクル中$\overline{AS}$がアサートにされっぱなしということはなくなりました。X68000のハードウェアは68000に合わせて設計されているので，元のところにはこの$\overline{AS}$と$\overline{RMC}$の論理和を出力するようにします。

### ●SIZ1，SIZ0，A0

68020では，ワード境界をまたぐワードアクセスが解禁となりました。しかも，ダイナミックバスサイジング機能により，8ビットバスへの接続も可能となりました。もうおわかりかと思いますが，68000にはなかったアドレスバスのA0が追加されたのです。これにともない，データストローブ信号の$\overline{UDS}$，$\overline{LDS}$は$\overline{DS}$1本に統一されました。また，SIZ1，SIZ0信号により，MPUがどのサイズのメモリアクセスを行っているかを知ることができるので，必要以上に広い幅のメモリがぶら下がっている場合でも，余計なデータを引っ掻くことはなくなりました。ちなみに，これらの信号の意味は表1に示したとおりです。

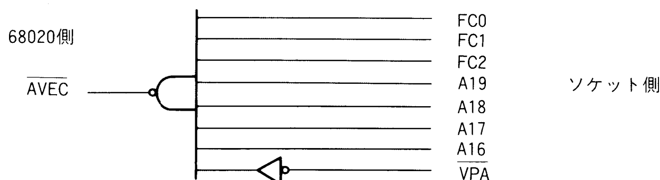
### ●$\overline{BGACK}$にともなう$\overline{BG}$の無効化

68000からEクロックを受け取ることにしました。また，スイッチひとつ（と，リセットスイッチかな？）で68000と68020の切り替えをできるようにしたい。となると，どちらかお休みになるほうのMPUを$\overline{BGACK}$をアサートして活動を中止してもらわなくてはなりません。しかし，68000には，このモードでもハイインピーダンスにならない出力信号が2つほどあります。ひとつは，前述のEクロック。残りは，MPUがこのモードにあることを示す$\overline{BG}$です。よって，この信号もMPUから，X68000内部へ垂れ流す前にひと捻り入れてやることにします。具体的には図3です。

＊　＊　＊

いきなり始めてしまった短期集中連載。文中では伏線を張りっぱなしで，いったいどこまでいくのかわからなかったかもしれませんが，実はこれを書いている本人にだってわからないのです。今後の最良のシナリオとしては，68EC030と68882を使ったアクセラレータを作って，プリント基板を起こして，どっかの会社から安価に販売してもらえればいいなぁ，とか思っているのですが，最悪のシナリオとして「世間をお騒がせしたことは私の不徳のいたすところであり，このような事態の……(中略)……誠に遺憾であります」といってうやむやになってしまうなどが考えられます。どこまでいけるのか，しばらくおつきあいください。

図2　AVECへの入力




## 今後の予定

本文中でもいったように，この連載は成功裏に終わるとは限りません。まだ，モノができているわけではないのですから。

とりあえず，製作が順調に進めば，だいたい以下のような予定で話を進めていこうと思います。

2月号　GALの焼き込み
3月号　ソフトウェア

とりあえず，このあたりまでには，アクセラレータの原形を作り上げようと思います。とにかく動くやつです。たぶん，ここまでではそんなに速度の向上は望めないでしょう。モトローラのアプリケーションマニュアルにも68020を16ビットバス幅固定で使用した場合は，同クロックの68000よりも遅くなると書いてあります。理由はいろいろとあるのですが，それはその都度お話しします。で，次になにをやるかはいろいろ考えているのですが，とりあえずMPUのクロックを本体よりも上げて（4月号かな？），以下のうちのどれかに手を出そうかと思っています。

**●68000と68020の動作切り替えスイッチを付け加える**

68020が68000の上位MPUとはいっても完全に上位互換性が保証されているわけではありません。詳しくはソフトウェア編で説明しますが，X68000の先輩格に当たるAMIGAやMacintosh用のアクセラレータには，その機能をキャンセルするスイッチがたいてい付いているそうです。すべてのプログラムが行儀よく書かれているとは限らないし，動作速度が変わってしまうと困るプログラムもいっぱいあるでしょう。そのたびにアクセラレータを抜いたり差したりするのは面倒なので，スイッチひとつで切り替えられるようにしておくとなにかと便利でしょう。

**●68881/2の接続**

やっぱ，68020といえばファンクションコードの中にCPU Spaceというのがあって，コプロセッサ用の命令なんかも用意されていたりするのです。MMU用のコプロセッサはとりあえずどっかに置いておいて，数値演算用のコプロセッサ68882や68881なんかとの接続も考えてみましょう。詳しくは9月号の瀧氏「FPP.MACの作成」を見てください。コプロセッサ専用の命令が用意されていると，どんなよいことが起こるのかが述べられています。

**●アクセラレータ内にメモリを載せて32ビット幅でメモリアクセスを行う**

68020は32ビット幅で使ったほうが本来の性能が出せるのです。

**●データバスが32ビット幅に移行できたらMPUに68030（もしくは68EC030）を使って……**

この先はお楽しみ。

**●製作が成功して読者からの要望がある程度たまれば，プリントパターンを起こして，どっかの町工場へ製作を依頼して，できるだけ安い値段で配布する**

などなど。

**参考文献**
1）中村仁一，ザ68020，プロセッサ1986年4〜11月号，技術評論社
2）細田誠，68000系ハードウェア設計ガイド，CQ出版社
3）辰野功，68020ファミリ活用の実際，電波新聞社

表1

| SIZ1 | SIZ0 | MPUがアクセスしようとするサイズ |
|---|---|---|
| 0 | 0 | ロング |
| 0 | 1 | バイト |
| 1 | 0 | ワード |
| 1 | 1 | 3バイト |

図3　$\overline{BG}$の無効化

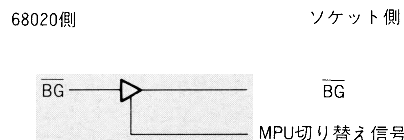
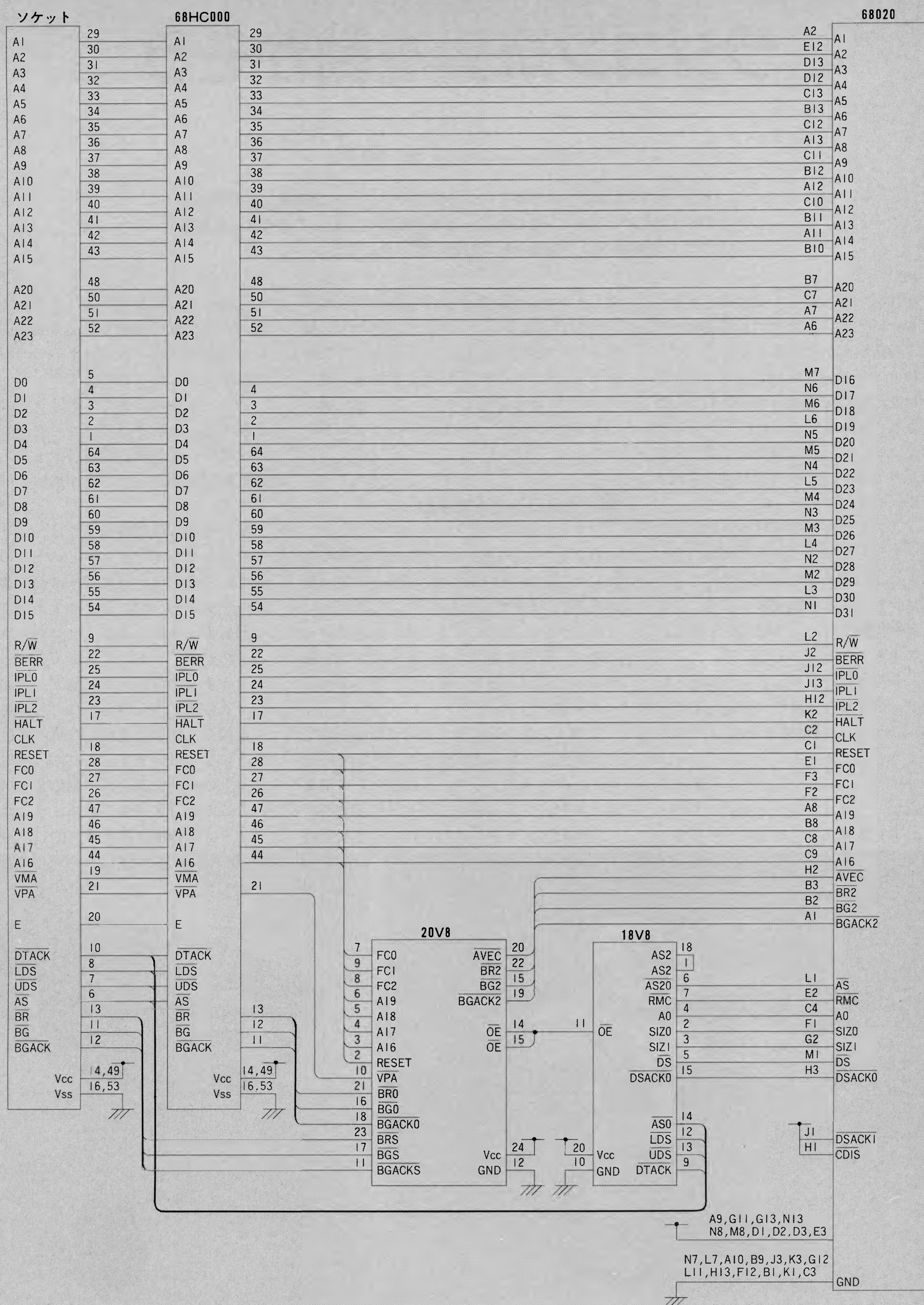


図4 回路図



ソケット
A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15
A20 A21 A22 A23
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14 D15
R/W BERR IPL0 IPL1 IPL2 HALT CLK RESET FC0 FC1 FC2 A19 A18 A17 A16 VMA VPA
E
DTACK LDS UDS AS BR BG BGACK
Vcc Vss
29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
48 50 51 52
5 4 3 2 1 64 63 62 61 60 59 58 57 56 55 54
9 22 25 24 23 17 18 28 27 26 47 46 45 44 19 21
20
10 8 7 6 13 11 12
14,49
16,53
68HC000
A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15
A20 A21 A22 A23
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14 D15
R/W BERR IPL0 IPL1 IPL2 HALT CLK RESET FC0 FC1 FC2 A19 A18 A17 A16 VMA VPA
E
DTACK LDS UDS AS BR BG BGACK
Vcc Vss
68020
A2 E12 D13 D12 C13 B13 C12 A13 C11 B12 A12 C10 B11 A11 B10
A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15
B7 C7 A7 A6
A20 A21 A22 A23
M7 N6 M6 L6 N5 M5 N4 L5 M4 N3 M3 L4 N2 M2 L3 N1
D16 D17 D18 D19 D20 D21 D22 D23 D24 D25 D26 D27 D28 D29 D30 D31
L2 J2 J12 J13 H12 K2 C2 C1 E1 F3 F2 A8 B8 C8 C9 H2 B3 B2 A1
R/W BERR IPL0 IPL1 IPL2 HALT CLK RESET FC0 FC1 FC2 A19 A18 A17 A16 AVEC BR2 BG2 BGACK2
L1 E2 C4 F1 G2 M1 H3
AS RMC A0 SIZ0 SIZ1 DS DSACK0
J1 H1
DSACK1 CDIS
GND
20V8
7 FC0
9 FC1
8 FC2
6 A19
5 A18
4 A17
3 A16
2 RESET
10 VPA
21 BR0
16 BG0
18 BGACK0
23 BRS
17 BGS
11 BGACKS
AVEC 20
BR2 22
BG2 15
BGACK2 19
OE 14
OE 15
Vcc 24
GND 12
18V8
AS2 18
AS2 1
AS20 6
RMC 7
A0 4
SIZ0 2
SIZ1 3
DS 5
DSACK0 15
11 OE
AS0 14
LDS 12
UDS 13
DTACK 9
20 Vcc
10 GND
A9,G11,G13,N13
N8,M8,D1,D2,D3,E3
N7,L7,A10,B9,J3,K3,G12
L11,H13,F12,B1,K1,C3

プロポ制御を行う

# ラジコン玩具を動かそう

Misawa Kazuhiko 三沢 和彦

コンピュータにつなぎたい玩具に対する読者アンケートの結果，リクエストNo.1だったのが「ラジコンをつなぐ」というものでした。ここでは，2種類のラジコン玩具をプリンタポートで制御してみましょう。

「皆さんはパソコンにどんなオモチャをつなげたらいいと思いますか？」最近はハードウェアの技術もかなり進んできて，プリンタを初めとして，スキャナ，ファクスなど，かなり高級な周辺機器が安価でつなげられるようになってきました。しかしながら，パソコンの可能性は無限大です。まだまだつなげて楽しいオモチャが考えられるはずです。

実は，この質問は1992年11月号の読者アンケートで出したもので，その回答のなかは「ラジコン」というものが圧倒的でした。なかにはラジコンを自動操縦にして，しかもカメラを搭載するというなんとも怪しげな？　アイデアもありました。

さて，今月はハードウェア工作特集ということで，皆さんのご要望にお応えして，X68000にラジコン送信機をつなげて楽しんでみようと思います。

X68000にラジコンをつなげる実験は以前粂野氏が「ラジコンスティックの製作」と題してすでに発表しています。そのときは，ラジコン受信機を設計自作してX68000につなげ，市販のラジコン送信機からの操作をアナログジョイスティックとして動作させるというものでした。今回は，逆にX68000側に送信機をつないで，X68000から制御命令を出して市販のラジコン玩具をコントロールさせることを考えます。

## ラジコンの仕組み

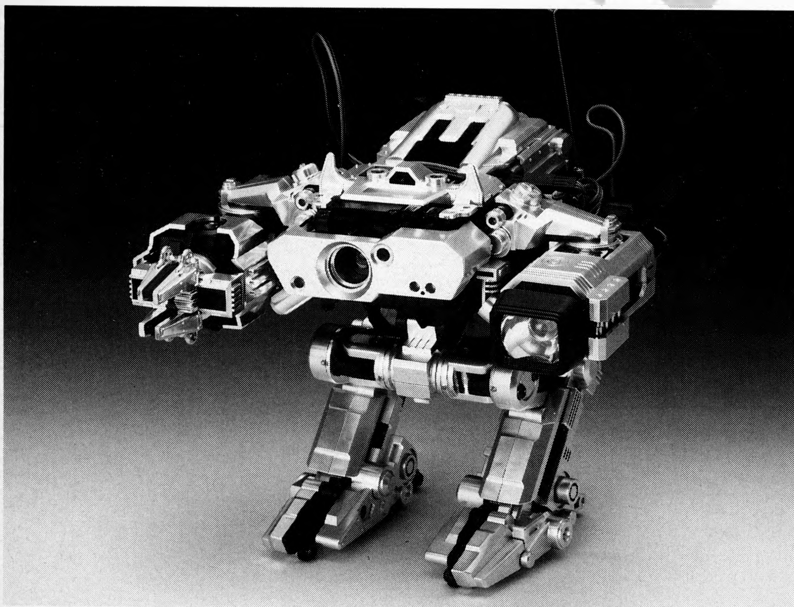
ラジコンは，一般的に周波数27MHzの電波にコントロールのための情報をAM（振幅変調）で載せて送信しています。AMというのはラジオ放送でもよく用いられている方法で，電波に載せる音声信号について音の大小で電波の強さを変えてやることによって情報を伝達します。

このラジコン送信機からの送信信号はすでに粂野氏によって解析されています。そのときの記事によると，送信機からの電波は基本的に一定の強さで出力されており，一定周期でデータが送信されるときに電波が途切れている箇所があります。4チャンネルの送信機では図1のように途切れる箇所が5箇所あり，それによって，パルス的な波形が4ブロックできます。

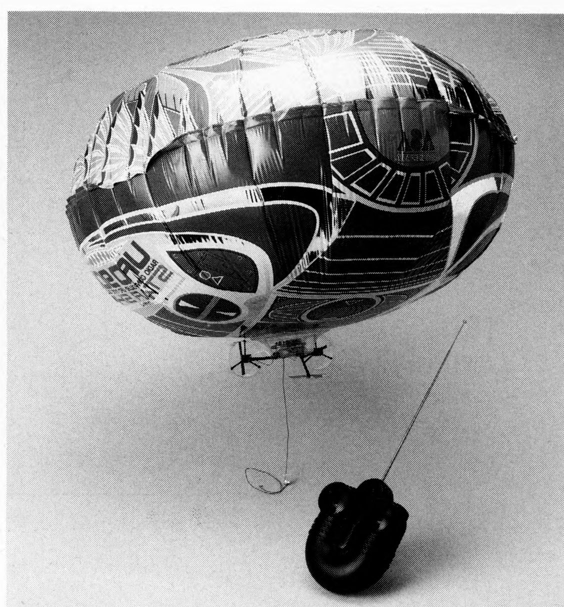
各チャンネルの操作スティックの傾きに比例して，それぞれのブロックの時間的幅が変化し，受信機側では各チャンネルの時間的幅を読み取って，操作情報を解読するのです。

操作される模型のほうでは，マイクロモーターかサーボモーターというものが使われています。マイクロモーターは普通の動力モーターを小型にし，スティックの傾きで回転方向と回転速度とをコントロールするものです。一方，サーボモーターは回転というよりは位置決めに使うもので，スティックの傾け方と自分自身の回転角度を常に比較して傾きに比例した角度で停止するようになっています。これで，舵取りやロボットの姿勢位置などを制御するのです。

ここで注目すべきなのは，ラジコンがスティックの傾きをアナログ的に処理してい



4チャンネルプロポによるロボット



ヘリウムを使った飛行船

るという点です。スティックの傾け方に比例した制御ができるために，細かい操作が可能になっています。このようにスティックの傾きに比例した制御のことを「プロポーショナル制御」と呼んでおり，このことからラジコン送信機のことを「プロポ」と省略して呼ぶことが多いのです。桒野氏はこのプロポーショナル制御に着目して，ラジコンを使ったアナログジョイスティックの製作を実験したのです。

## ラジコン模型の製作

今回製作したラジコンは2機種あります。まずは，ヘリウムガスを使った気球に左右2機のプロペラが付いているものです。送信機には2チャンネルのコントローラが付いており，各チャンネルがそれぞれ片方のプロペラに対応しています。この2チャンネルを操作することにより，次のような動作が可能です。

前進：両翼正転
右折：左翼のみ正転
左折：右翼のみ正転
後進：両翼逆転
右折後進：左翼のみ逆転
左折後進：右翼のみ逆転

また，推進機の取り付け位置を工夫すれば，前進と上昇，後進と下降を組にすることもできます。

これらの動作はプロペラのON/OFFのみの制御で，プロペラの回転速度が連続的に変わるようにはなっていません。その意味では上で述べたプロポーショナル制御にはなっておらず，簡易型のコントローラになっています。

次に製作したのはこの気球よりはるかに複雑なラジコンロボットです。しかもこれはプラモデルになっていて，バラのプラスチック部品をねじ止めで組み立てていくという凝った模型です。編集部で私と(U)氏と（ふ）嬢との3人がかりで製作して，2時間くらいかかりました。このラジコンロボットには4チャンネルのコントローラがあって，

チャンネル1：前進・後進
チャンネル2：腰の左右運動
チャンネル3：首の前後運動
チャンネル4：右手のつかむ動作・左手のライト点灯

というように各チャンネルごとにそれぞれ2通りの動作ができるようになっています。しかも，各チャンネルがすべてプロポーショナル制御できます。すなわち，前後進の速度や腰の左右運動の角度などが連続的に変えられるということです。

では，これからそれらのラジコンをX68000につなげていく実験を順番に説明していきましょう。

## 簡易型ラジコン送信機

ひとつめのラジコン送信機の中身を開けてみると，各チャンネルが2接点のスイッチに対応していて，図2のようにどちらかの端子に可動接点を接触させるだけのものになっていました。このコントローラはもともとデジタル制御なので，これをX68000から制御することはきわめて簡単です。図3-1のようにスイッチの端子に番号を付けて，

SW1(a)：右翼正転
SW1(b)：右翼逆転
SW1(c)：右翼停止
SW2(a)：左翼正転

**図1　プロポの送信波形**

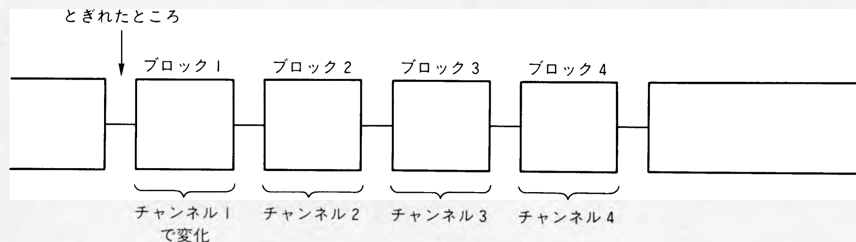



**図2　簡易型コントローラのスイッチ**

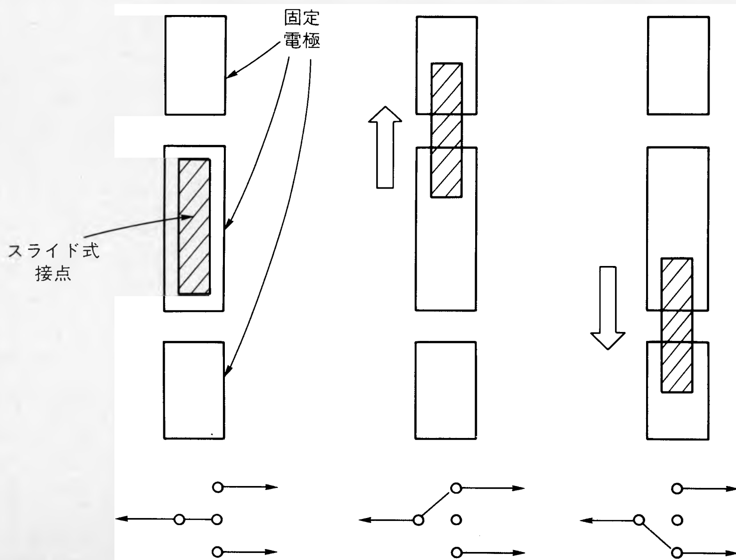



**図3-1　元のスイッチのパターン**

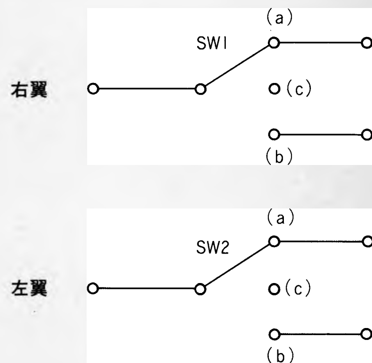
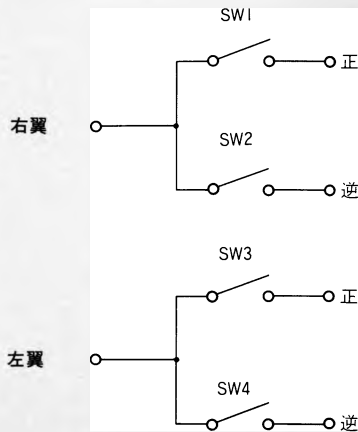



**図3-2　改造したスイッチパターン**



SW2(b)：左翼逆転

SW2(c)：左翼停止

となっているところを図3-2のように作り替えて，

SW1ON　SW2OFF　：右翼正転
SW1OFF　SW2ON　：右翼逆転
SW1OFF　SW2OFF　：右翼停止
SW3ON　SW4OFF　：左翼正転
SW3OFF　SW4ON　：左翼逆転
SW3OFF　SW4OFF　：左翼停止

とします。このように作り替える理由は，回路の都合上，接点選択型のスイッチよりは単にON/OFF型のスイッチのほうが多少配線が増えるように見えても全体として簡単になるからです。

このときの論理を表にすると，表1のようになります。そこで，SW1～4の4つのスイッチのON/OFFを2進数の各論理ビットの1/0に置き換えることによって，デジタル命令で表現することができるようになるのです。

今回ラジコンをコンピュータ制御にするわけですが，送信機自体を作り直すのは，Oh!Xの読者のうちでも相当ハードウェアの経験のある人でないときついかもしれません。そこで，送信機の大部分の回路は流用し，最小限の改造を加えることで済ませたいと思います。そこで，このスイッチ部分だけデジタル回路に置き換えることを考えます。

表1

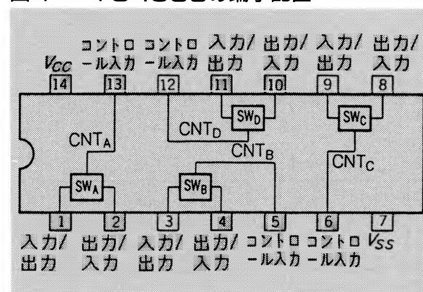
| | | SW1 | |
|---|---|---|---|
| | | ON | OFF |
| SW2 | ON | ╳ | 逆転 |
| | OFF | 正転 | 停止 |

## アナログスイッチ

外部からのデジタル信号でスイッチのON/OFFを切り替える回路に「アナログスイッチ」というものがあります。アナログスイッチの場合は，機械的接点を持っておらず，トランジスタを使った電子素子で電流のON/OFFを行います。

アナログスイッチで代表的なのは，TC4066というICです。このICはCMOSでできており，論理回路でよく使われているTTLICと違って電源は3～18Vと幅広く使えるのが特徴的です。TC4066の端子配置を図4に載せます。このパッケージにはH/Lのコントロール信号によってON/OFFが制御できるスイッチが4個入っています。ひとつのスイッチにはそれぞれ2つの端子が付いていますが，その2つのあいだに電気的な区別はありません。

図4　TC4066の端子配置

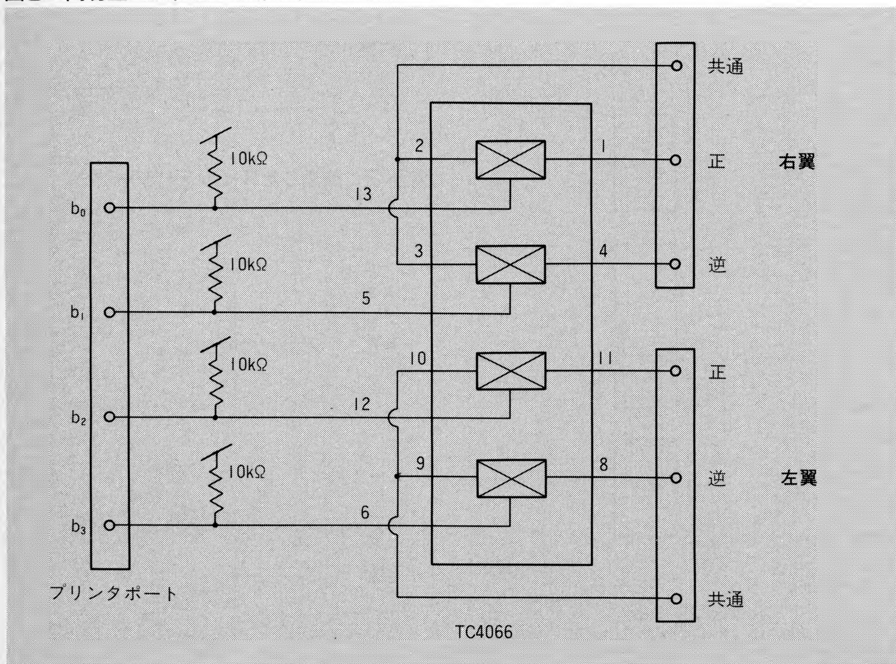



アナログスイッチは機械式接点のスイッチとほぼ置き換えることができますが，電子的なスイッチである以上，いろいろな制約があります。初歩的な使い方の範囲内で注意すべきなのは，ON状態でも端子間に一定の抵抗値が残っているという点と，端子間に電源電圧以上の電圧はかけられないという点です。

このアナログスイッチTC4066を上で述べたラジコン気球の制御に使います。回路図は図5に示しますが，IC1個だけの簡単な回路で，構成は図3-2とまったく変わりないことがわかると思います。アナログスイッチの動作電圧が電源電圧以下ということから，操作すべきコントローラの電源とこのインタフェイスと共通に使うことにします。

X68000との接続に関しては，2チャンネル合わせて4ビットパラレルの出力が必要ということなので，出力が3ビットしかとれないジョイスティックポートは使えません。今回はあとで6ビットパラレル出力が必要となることもあって，プリンタ出力ポートを使うことにします。回路中の10kΩ抵抗についてはプリンタポートとのインタフェイスに必要なのですが，それについて次に説明します。

図5　簡易型コントローラインタフェイス




## X68000プリンタ出力ポート

X68000のプリンタ出力はセントロニクス仕様というプリンタ各機種に共通のインタフェイスを使っています。セントロニクスの詳細は省略することにして，ハードウエア的な観点から説明するために，図6を見てください。特に今回有用なのは8ビットパラレル出力のLS273とLS07の部分です。LS273は8ビットパラレルDフリップフロップで，現在のハードウェア工作入門の連載のほうでも扱っているレジスタの役割をしています。これは，CPUから出力すべきデータをセットしたら，次にセットするまでデータを保持してくれるものです。

LS07は出力バッファといって，TTL規格の5V以外の外部機器も接続できるようにレベル変換を行えるものです。これは特にLS07の出力がオープンコレクタで，かつ耐圧が30Vまであるということによります。

通常TTLレベルで5V以上の論理回路をON/OFFするにはトランジスタスイッチ

を使います。トランジスタスイッチは以前ハードウェア工作入門の連載でいろいろな回路に応用したものですが，図7-1のような回路になっています。入力のベース端子がHレベルになるとベースからエミッタに電流が流れ，コレクタに外部電源をかけておくとそれによってコレクタからエミッタにも電流が流れるという仕組みです。

このときベースにかけるHレベルとコレクタにかける電源電圧とを変えることができるので，レベル変換に使えるというわけです。オープンコレクタというのはICの内部で出力段にレベル変換用のトランジスタスイッチが付いており，図7-2のようにコレクタ端子が開放されている状態になっています。そこで，抵抗を通して外部電源に接続するだけでH/LのレベルがTTL入力の5Vから外部電源の電圧に変換できるのです。

このLS07のおかげで，外付け抵抗だけでX68000内部の5V論理回路から，今回のように5V以上のラジコンコントローラを駆動することができるのです。

## プロポーショナル制御のラジコン

次にラジコンロボットを操作するプロポーショナル制御のラジコン送信機をX68000につなぐことを考えたいと思います。

ラジコンのプロポの中身を開けてみると，スティック部分にはボリュームが取り付けられており，ボリュームの回転角がスティックの傾きを表すようになっています。そして，送信機はこのボリュームの抵抗値を読み取り，それに比例する信号を電波で送って，制御のための情報とします。ラジコンのプロポだけではなく，アナログジョイスティックの位置決めはたいていこのボリューム方式を採用しています。

プロポで制御するとき，直接操作するのはこのボリュームですから，ボリューム自体をデジタル制御できるようなものにそっくり入れ替えることができればX68000から操作できるということになります。しかもボリュームにある3端子の配線をつなぎ変えることだけで対処できるので，改造も最低限で済みます。

## デジタル制御式電子ボリューム

通常のボリューム（可変抵抗器）の仕組みは図8に示すとおりで，円弧状の抵抗板（固定抵抗器）に機械式の可動接点があります。3つある端子のうち両端の2つはそれぞれ固定抵抗器の両端についているのでそれらのあいだの抵抗値は常に変わりません。

真ん中の端子がスライド式の機械式接点についており，ボリュームを回すと接点の接触している位置が連続的に変わります。真ん中の端子と両側のどちらかの端子との間の抵抗値は可動接点と端の端子までの距離に比例しています。

このボリュームをデジタル制御にすることを考えましょう。それには，本来連続量（アナログ量）である可動接点の位置を飛び飛びの値に区切ってやることにします。飛び飛びのひとつの単位は決まった量にしておくとすると，このときのボリュームは図9のように1単位分の固定抵抗を直列につなげて，それぞれのあいだに分割端子を付け，ボリュームの真ん中の端子はそれらの分割端子のどれかひとつを選ぶということにすればよいのです。

分割端子のどれかひとつを選ぶスイッチといえば，それは先ほど述べたアナログスイッチの応用となります。しかしながら今回の場合は端子2つのあいだのON/OFFではなくて，複数の端子からひとつの端子を選択するタイプのものが必要となります。

多接点型のスイッチの切り替えをデジタ

図6　プリンタインタフェイスブロック図

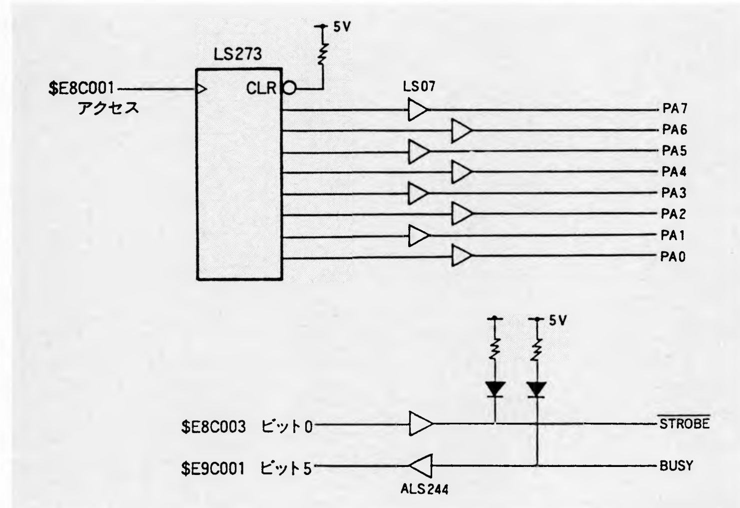



図7-1　トランジスタスイッチ

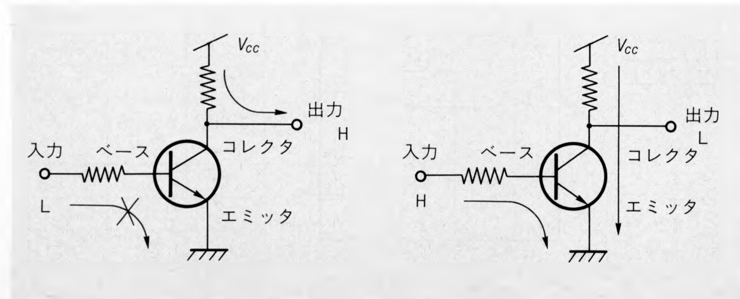



図8　ボリューム（可変抵抗器）のしくみ

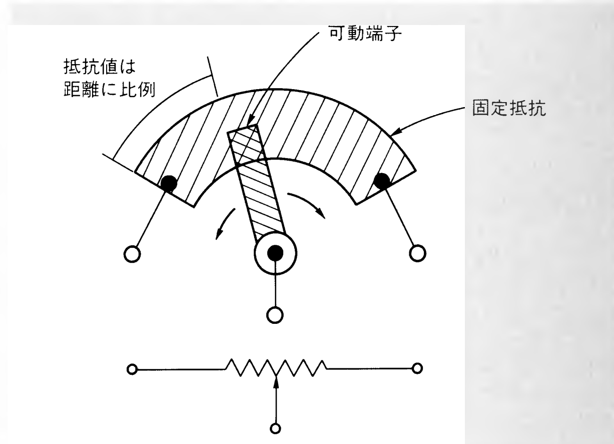
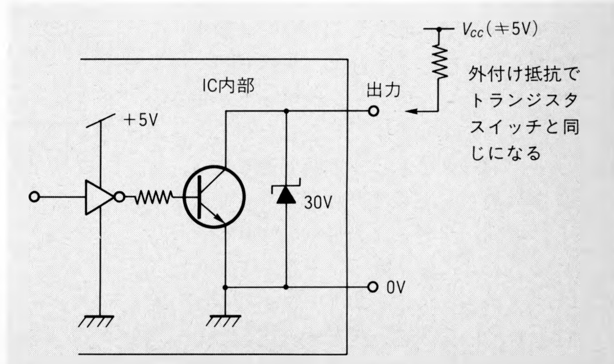



図7-2　オープンコレクタ



ル信号で行うのにちょうどよいICパッケージがTC4051/4052/4053のシリーズです。これらの品種の接点構成と端子配置図とを図10に示します。TC4051が1回路8接点，4052が2回路4接点，4053が3回路2接点のスイッチ構成です。今回はTC4051を使用するので，4051を中心に説明を続けたいと思います。

4051はひとつの共通接点と8つの選択接点を持っています。このように共通接点と多接点とのあいだで選択を行うようなスイッチを「マルチプレクサ」と呼んでいます。接点の選択は3ビットのコントロール端子に論理信号を入れることによって行います。その他，すべての接点との接続を切るINH端子があります。電源系統に関しては，プラス側がコントロール信号のHレベルとスイッチにかける電圧とで共通になっていますが，マイナス側はコントロール信号のLレベルとスイッチにかける電圧とで別々になっています。これは，論理回路は0-Vcc間でH/Lを決めるのに対して，スイッチには正負の反転する交流信号をかけることができるようになっているためです。

このアナログマルチプレクサを使って今回は図11のような回路を組むことにします。このとき，図13のような抵抗値に設定すれば，それぞれの分割端子における可変抵抗値は表のように対応することになります。こうして，デジタル信号で抵抗値が制御できる電子制御ボリュームを組み立てることができるのです。

## 4チャンネルプロポへの応用とX68000の接続

ラジコンロボットは4チャンネル制御なので，上で述べた電子制御ボリュームが4個必要となってきます。この4チャンネルをデジタル制御することになるのですが，1チャンネルあたり3ビット必要なので，全部で3×4＝12ビット必要ということになります。

X68000とのインタフェイスを考えるとき，拡張ボードを自作することなしに簡単に外部機器をつなぐとすれば，先ほど述べた8ビットパラレルのプリンタポートが最適ということになります。しかしながら，12ビットのデータを8ビット幅で送るには，データを分割して送らなければなりません。しかも，分割したデータの最初のブロックを送ったあと，次のブロックを送るときにはプリンタポートには先のデータが残っていませんから，先のデータを読み出した外部機器の側で保持しておかなければなりません。

データの保持には最近お馴染みのDフリップフロップを使います。このとき，3ビットのデータを4チャンネル独立に保持できるようにしておくと便利です。今回はTTL規格表から4ビットパラレルDフリップフロップのHC175を選んでみました（表2）。電源電圧を5Vに限らないようにするためにCMOSタイプのHCシリーズを使います。

Dフリップフロップは入力端子にデータを入れた状態で，クロック端子にLレベル

図9　ボリュームのデジタル化

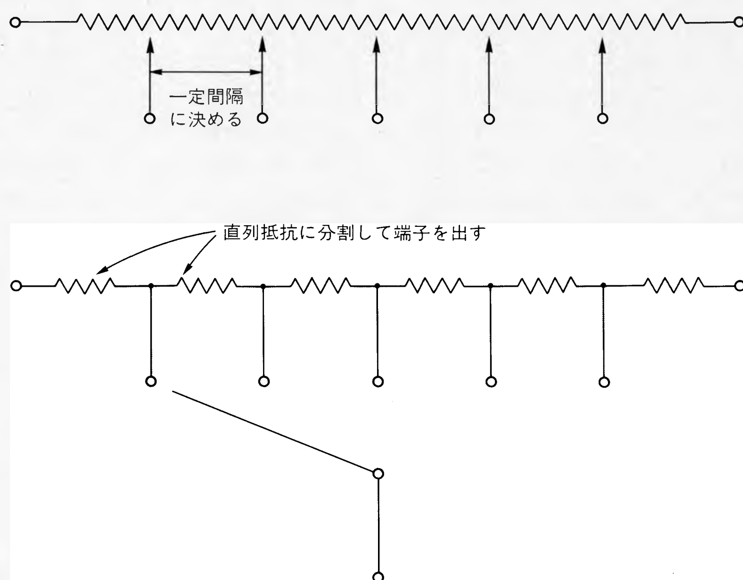
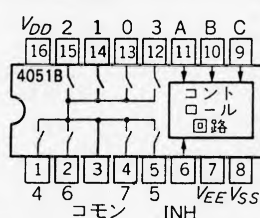
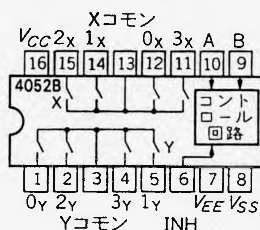
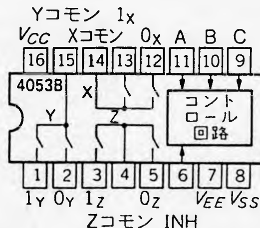



図10

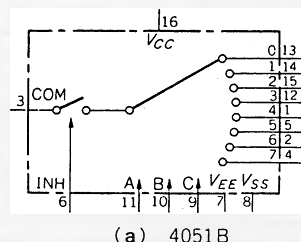
4051B/4052B/4053Bの端子配置



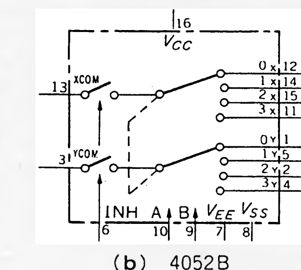



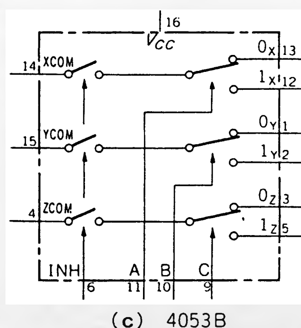

接点構成




（a）　4051B




（b）　4052B




（c）　4053B

| 入力状態 | | | | "ON"チャネル |
|---|---|---|---|---|
| INHIBIT | C | B | A | |
| 4051B | | | | |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 2 |
| 0 | 0 | 1 | 1 | 3 |
| 0 | 1 | 0 | 0 | 4 |
| 0 | 1 | 0 | 1 | 5 |
| 0 | 1 | 1 | 0 | 6 |
| 0 | 1 | 1 | 1 | 7 |
| 1 | X | X | X | —— |
| 4052B | | | | |
| INHIBIT | | B | A | |
| 0 | | 0 | 0 | $0_X,0_Y$ |
| 0 | | 0 | 1 | $1_X,1_Y$ |
| 0 | | 1 | 0 | $2_X,2_Y$ |
| 0 | | 1 | 1 | $3_X,3_Y$ |
| 1 | | X | X | —— |
| 4053B | | | | |
| INHIBIT | A or B or C | | | |
| 0 | 0 | | | $0_X,0_Y,0_Z$ |
| 0 | 1 | | | $1_X,1_Y,1_Z$ |
| 1 | X | | | —— |

X＝Don't care

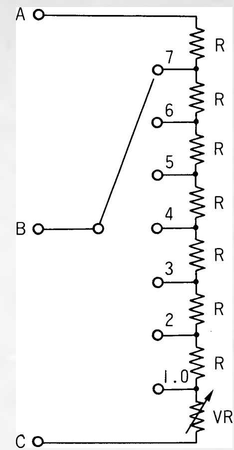
からHレベルに立ち上がる信号を入れると出力端子の出力が入力データにロックされ，そのあと次にクロック信号を入れるまでは入力データが変動しても出力は保持されたままになっています。そこで，4チャンネルの制御を行うためには，X68000からの出力を4チャンネルすべてのDフリップフロップの入力にかけておいて，保持したいチャンネルのクロック信号だけあとから加えてやるというかたちにします（図14）。

次にデータを保持したいチャンネルの選択を行うための回路について考えます。それには図15のように4チャンネルそれぞれにクロック出力をつないでおき，そのクロック信号を目的のチャンネルに振り分ければよいことになります。

クロック信号が伝わらないチャンネルにはデータの変化は伝わらないので，出力は一定のままになっています。図を見ると一見アナログスイッチと同じ回路のように見えますが，入出力のレベルがH/Lで一定であるデジタル論理回路では，アナログスイッチよりももっと簡単な回路で信号の振り分けが可能です。

このように，ひとつの出力を目的の出力に振り分けるための論理回路を一般的に「デコーダ」と呼びます。デコーダICには8チャンネル1回路のHC138と4チャンネル2回路のHC139とがありますが，今回は4チャンネル用のHC139を使います。HC139の規格は表3のとおりです。これには出力端子が4つあり，その4つの出力端子を選択する2ビットのセレクト端子があります。セレクト入力で選んだ出力だけをLレベルにし，ほかはHレベルのままになっています。このとき，イネーブル端子がLになっていなければならず，イネーブル端子をHにしておくと，すべての出力がH

図13




R＝620Ω
VR＝1kΩ→620Ωに設定したとする

| 端子 | AB間 | BC間 | AC間 |
|---|---|---|---|
| 7 | 620Ω | 4340Ω | |
| 6 | 1240Ω | 3720Ω | |
| 5 | 1860Ω | 3100Ω | |
| 4 | 2480Ω | 2480Ω | 4960Ω |
| 3 | 3100Ω | 1860Ω | |
| 2 | 3720Ω | 1240Ω | |
| 1 | 4340Ω | 620Ω | |

図11　電子制御ボリューム

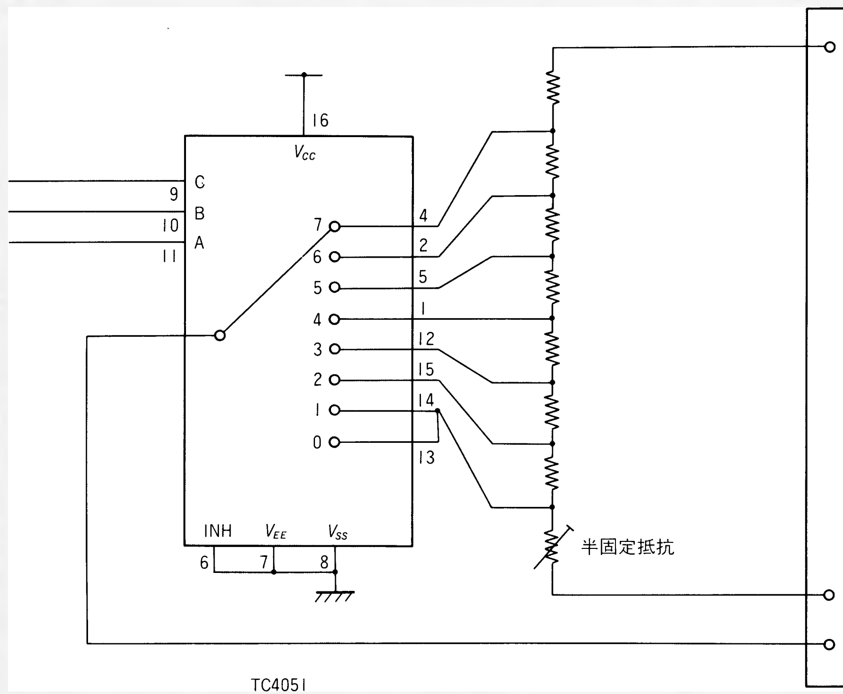



図14

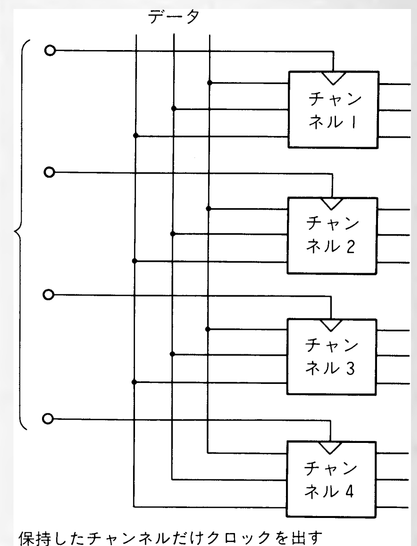
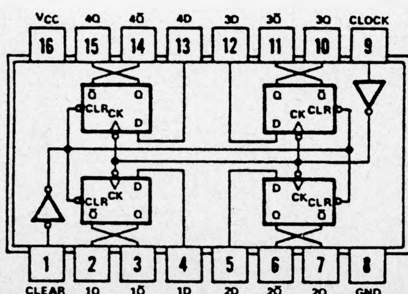
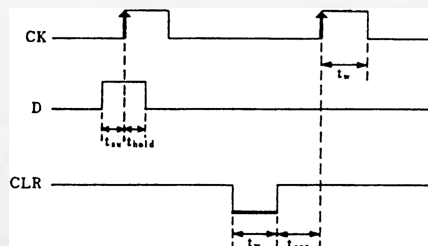
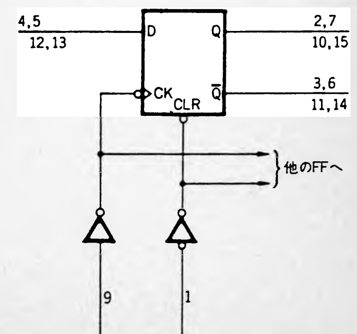



図12-1　HC175



になってしまいます。そこで，イネーブル端子をクロック入力にしてL→H→Lという信号を入れてやると，セレクトで選んだ出力だけにL→H→Lのクロック信号が出てくることになります。

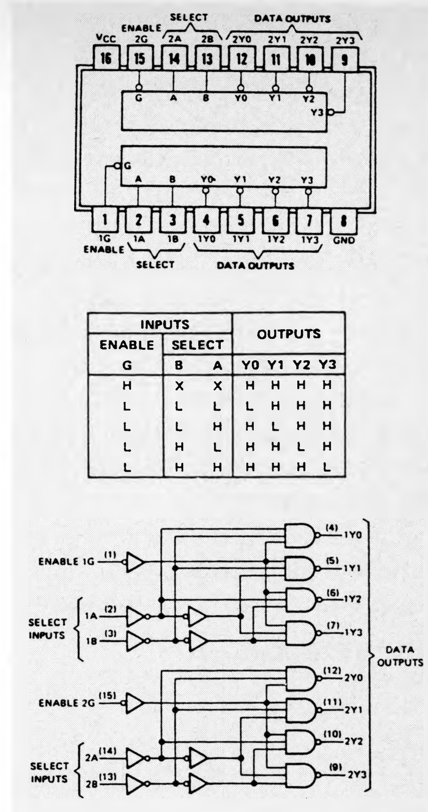
以上のフリップフロップへの入力データ3ビット，チャンネルセレクト2ビット，クロック信号1ビットの全部で6ビットの信号線で4チャンネル電子ボリューム回路をコントロールすることができるわけなのです。全体の回路図は図16にまとめてみました。

## 製作

簡易型コントローラのインタフェイスのほうはICが1個だけですから，実体配線図を示すだけで，説明の必要はないと思います（図17）。

4チャンネル電子ボリュームのほうは実

図12-2 HC139




| INPUTS | | | OUTPUTS | | | |
|---|---|---|---|---|---|---|
| ENABLE | SELECT | | | | | |
| G | B | A | Y0 | Y1 | Y2 | Y3 |
| H | X | X | H | H | H | H |
| L | L | L | L | H | H | H |
| L | L | H | H | L | H | H |
| L | H | L | H | H | L | H |
| L | H | H | H | H | H | L |

図15 クロックのチャンネル振り分け

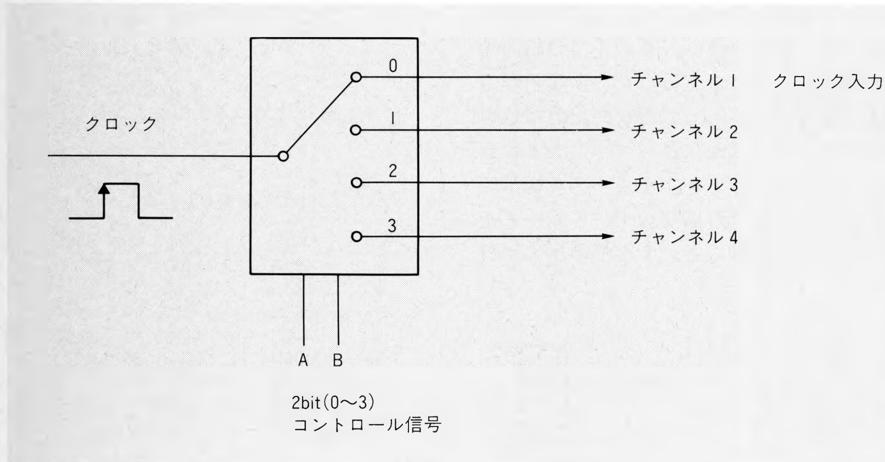



図16 4チャンネルプロポーショナルコントローラ インタフェイス回路図

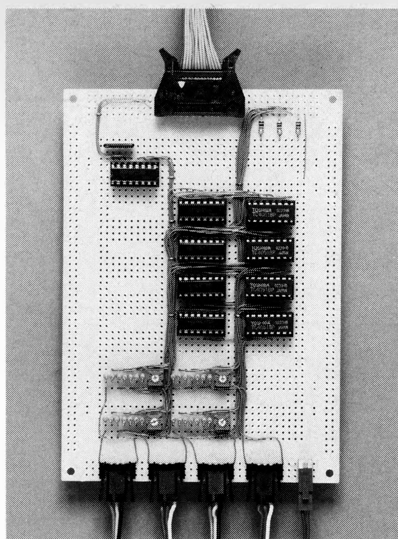
○各ICの$V_{cc}$，GND端子を忘れないで配線する。
○各ICの使わない入力端子はGNDに直結する。

同上(図12参照)

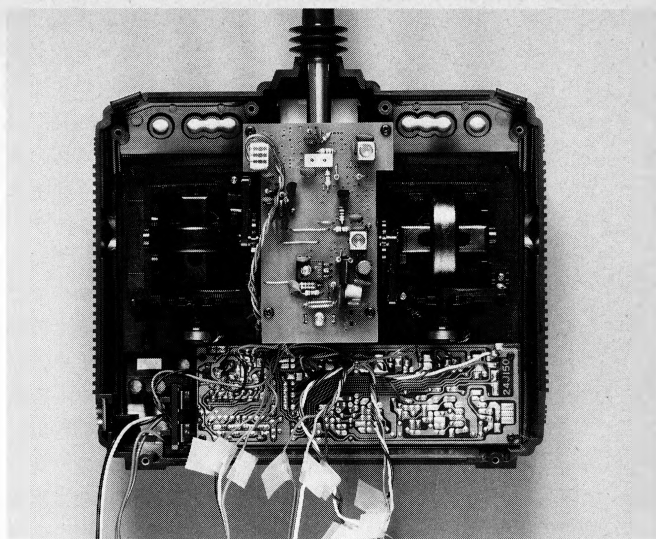
HC157×4　TC4051×4

体配線といってもほとんどすべてがジャンパ線です。非常に混乱しやすいので，ゆっくり確認しながら配線していきます。といっても，1チャンネル分の回路はそんなに難しいわけではなく，4チャンネル同じ回路を組み立てるのが全体の配線量を多くしているだけです。

部品の配置はチャンネルごと，ブロックごとにまとめていきます。実体配線図はあまりに込み入るので省略しますが，今回私が製作した基板の部品配置だけ参考のため図18に示しておきます。



できあがった4チャンネルプロポ型



プロポ側への配線

## プリンタポートデータ出力用外部関数

今回設計製作したラジコンコントローラインタフェイスを動かすのに，X68000のプリンタポートからデータを出力する外部関数をプログラムしてみました。リスト1がそのソースリストです。

外部関数prtout(data)はプリンタポートにパラレル8ビットデータを出力するだけのものです。X68000のプリンタポートは図19のような端子からなっていて，このprtout関数は端子NO.2～9のPA0～7だけ使います。このコネクタはヒロセムセンというメーカーの14ピンフラットケーブルコネクタがそのまま使えるので，ほかの自作ボードを接続するときにも一般的に応用することができます。

この関数も今回に限らずプリンタポートを使った8ビットパラレルデータの出力には汎用として使えるものになっています。外部関数はインタプリタ/コンパイラ共用になっています。

CC prtdrv.s

としてソースリストをコンパイルし，生成されたprtdrv.xのファイル名をprtdrv.fncとしてX-BASICのあるディレクトリと同じところに入れます。BASIC.FNCファイルに

func = prtdrv

と加えて起動すれば，この関数はprtout(data)の命令形でプリンタポートの8ビットパラレル出力にdata値をセットすることができます。

この外部関数で簡易型コントローラと4チャンネルプロポの両方が制御できます。

## 動作チェックとプログラム

全体の配線が終わったら，プロポに接続する前に動作チェックです。簡易型コントローラのほうは動作チェックの必要もないほど簡単な回路なので，以下説明は省略して，ここでは，4チャンネルプロポのほうに絞って説明していきたいと思います。各チャンネルの3端子のうち両端の端子のど

図17　簡易型インタフェイス

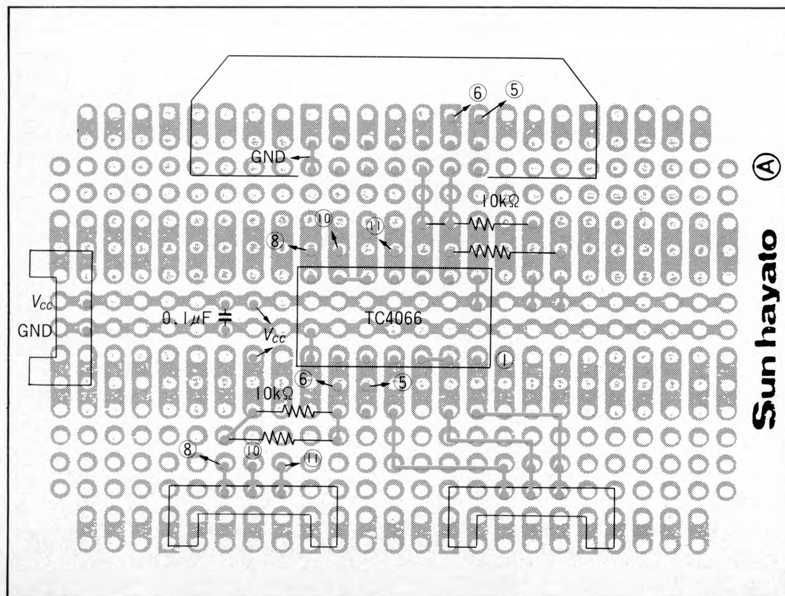
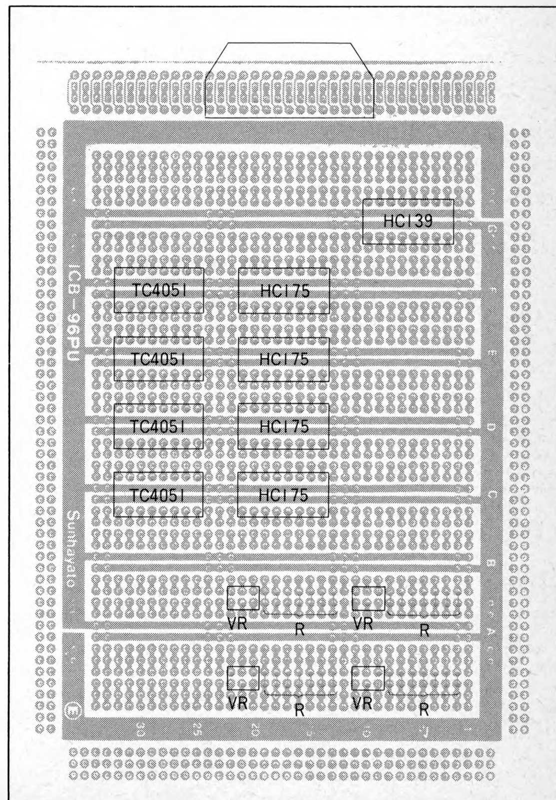



図18　4チャンネルプロポ型の部品配置例





これで接続完了

ちらか一方と真ん中の端子とのあいだの抵抗値をテスターで計りながら，リスト1の外部関数を組み込んだX-BASICを立ち上げ，リスト2のサンプルプログラムを打ち込んで走らせます。

プログラムでは初期設定として各チャンネルともボリュームがセンターにくるようになっています。キーボード上から前後左右の入力キー，

| | | | |
|---|---|---|---|
| | E | | I |
| S D | | M K | |
| | X | | J |

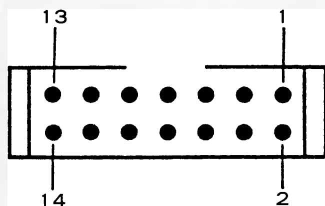
を順番に押していって，テスターの表示が変わるのが確認できたら大成功です。このとき，いま測定しているチャンネルと違うところのキーを押しても抵抗値が変化しないことも確認してください。

プロポにつなぐ前に基板上で電子ボリュームのコントロールができれば，あとはボリュームの3端子を間違いなくつなぐだけでOKです。プロポをつないでプロポとインタフェイスの共通電源をつなぎ，プログラムを走らせます。

このとき，ラジコンロボット本体の電源はまだ切っておきます。基板上の半固定抵抗4個をすべてだいたい真ん中の位置にすると，すべての動作を停止させていることに対応します。しかしながら，ここでラジコンロボットの電源を入れると，たいていの場合は動き出してしまいます。これはボリュームのバランスがずれているためで，その補正を半固定抵抗で行います。4チャンネルすべての半固定抵抗を調整して，動作が停止するか，あるいは姿勢が正立位置にくるようにします。

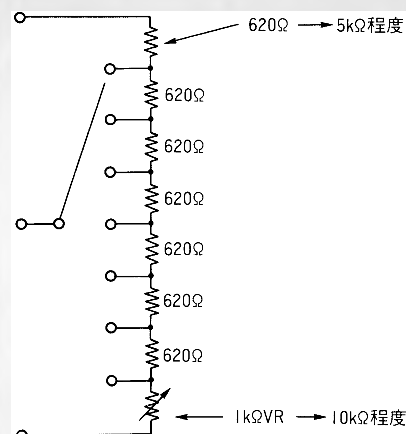
実際にキーボード上から前後左右の入力キーで操作してみると，設計どおり動きだしましたので，ひと安心でした。しかしながら，実際に動作させてみるとどうも分割抵抗の値の設定がうまくいかなかったらしく，きちんとプロポーショナル動作してくれません。この問題を解決するには，分割抵抗の部分の印をつけた抵抗値（図20）を変えてみるのがよいでしょう。

**図19 プリンタ用コネクタ**




| 端子No. | 信号名 | I/O | 備 考 トロー |
|---|---|---|---|
| 1 | $\overline{\mathrm{STROBE}}$ | Out | 負極性のプリンタに出力するライトストローブ信号 |
| 2 | PA0 | Out | パラレルデータ |
| 3 | PA1 | 〃 | 〃 |
| 4 | PA2 | 〃 | 〃 |
| 5 | PA3 | 〃 | 〃 |
| 6 | PA4 | 〃 | 〃 |
| 7 | PA5 | 〃 | 〃 |
| 8 | PA6 | 〃 | 〃 |
| 9 | PA7 | 〃 | 〃 |
| 10 | N.C | — | 非接続 |
| 11 | BUSY | In | プリンタがレディ状態のとき"LOW"レベルになる |
| 12 | N.C | — | 非接続 |
| 13 | GND | — | グランド |
| 14 | GND | — | グランド |

**図20 電子ボリュームの改良**



両端の抵抗を10kΩ半固定抵抗にして調整してみるのもよい

**表4 部品価格表**

| | | |
|---|---|---|
| **●接続ケーブル** | | |
| 14ピン圧着コネクタ | 2個 | @90円 |
| 14ピンフラットケーブル | 1m | 140円 |
| **●簡易型コントローラインタフェイス** | | |
| TC4066 | 1個 | 230円 |
| 14ピンICソケット | 1個 | 30円 |
| IC用基板(サンハヤトICB-86) | 1枚 | 90円 |
| 14ピンコネクタヘッド(HIF-3BA-14PA-DS) | 1個 | 115円 |
| 2ピンコネクタ(HNC2.25S-3) | 1個 | 10円 |
| 2ピン基板ターミナル(HNC2.25P-2-DSL) | 1個 | 15円 |
| 3ピンコネクタ(HNC2.25S-3) | 2個 | @10円 |
| 3ピン基板ターミナル(HNC2.25P-3-DSL) | 2個 | @15円 |
| コンタクトピン | 8本 | @5円 |
| 10kΩ抵抗 | 4本 | |
| 0.1μFコンデンサ | 1個 | |
| スズメッキ線 | 少々 | |
| ビニール配線材 | 少々 | |
| **●4チャンネルプロポインタフェイス** | | |
| TC4051 | 4個 | @80円 |
| HC139 | 1個 | 60円 |
| HC175 | 4個 | @60円 |
| 16ピンICソケット | 9個 | @30円 |
| IC用基板(サンハヤトICB-96PU) | 1枚 | 535円 |
| 14ピンコネクタヘッド(HIF-3BA-14PA-DS) | 1個 | 115円 |
| 2ピンコネクタ(HNC2.25S-2) | 1個 | 10円 |
| 2ピン基板ターミナル(HNC2.25P-2-DSL) | 1個 | 15円 |
| 3ピンコネクタ(HNC2.25S-3) | 4個 | @10円 |
| 3ピン基板ターミナル(HNC2.25P-3-DSL) | 4個 | @15円 |
| コンタクトピン | 14本 | @5円 |
| 10kΩ抵抗4本アレー | 2個 | @50円 |
| 620Ω抵抗 | 28本 | |
| 1kΩ半固定抵抗 | 4個 | @90円 |
| 0.1μFコンデンサ | 9個 | |
| スズメッキ線 | 少々 | |
| ビニール配線材 | 少々 | |
| **●ラジコン機材** | | |
| TOMY TXR-002 | | 39,800円 |
| AIR SHIP ASIA UFO-900　STARSHIP TYPE-1 | | 19,800円 |

## 最後に

いかがでしたか？　サンプルプログラムのようにただキーボード上から操作させるだけではパソコンからコントロールする意味がありません。たとえば操作した手順をリアルタイムでコンピュータのメモリに書き込んでいくようにして，まったく同じ動作を再現させてみるという手もあります。またセンサーを付けてまわりの障害物を常時監視し，障害物をよけながら目的地に到達させるようなプログラムも可能です。しかしながら，そのときはラジコン送信機をもう１台用意して模型のほうにも搭載しなければなりませんから，回路がさらに複雑になります。しかしそれができれば，それこそ隠しカメラを載せてラジコンでスパイをさせるなどというアドベンチャー映画に出てきそうな芸当も可能でしょう。それはまたの機会ということにします。

**リスト1**

```
****************************************************
*
*       ハードウェア工作入門   I/Oドライバ
*              外部関数  prtout
*
*          Ver. 1.0     1992.11.23  K. Misawa
*
*            prtout(data) : データ出力
*                           (引数)  データ
*                           (戻り値) なし
*
*          X-BASIC外部関数
*            & BASTOCライブラリ   兼用
*
****************************************************

                .include        doscall.mac
                .include        fdef.h
                .globl          _prtout
                .text
                .even

****************************************************
*
*               外部関数ヘッダ部
*
****************************************************

*
*インフォメーションテーブル
*
                dc.l            x_init
                dc.l            x_run
                dc.l            x_end
                dc.l            x_sys
                dc.l            x_brk
                dc.l            x_ctrl_d
                dc.l            x_res1
                dc.l            x_res2
                dc.l            ptr_token
                dc.l            ptr_param
                dc.l            ptr_exec
                dc.l            0,0,0,0,0

x_init:
x_run:
x_end:
x_sys:
x_brk:
x_ctrl_d:
x_res1:
x_res2:

                rts
*
*関数名テーブル
*
ptr_token:      dc.b    'prtout',0
                dc.b    0

                .even

*
*パラメータテーブル
*
ptr_param:      dc.l    prtout_par

*
*パラメータIDテーブル
*
prtout_par:     dc.w    int_val
                dc.w    void_ret

*
*実行アドレステーブル
*
ptr_exec:       dc.l    prtout_exec

                .even

****************************************************
*
*                    定義関数ルーチン
*
****************************************************


*
*データ出力関数  prtout(data)
*
***ポートアドレス
port            equ     $e8c001

***実行アドレス
prtout_exec:

***引数をスタックに積み替え
                move.l  12(sp),d1
                move.l  d1,-(sp)

                bsr     _prtout
                addq.l  #4,sp

                move.l  #0,d0
                rts

***メインルーチン
_prtout:

***スーパバイザモードに入る
                clr.l   -(sp)
                dc.w    _SUPER
                addq.l  #4,sp
                move.l  d0,spbuf

***プリンタポートへ書き込み
wr_ok:          move.l  #port,d2
                movea.l d2,a3
                move.l  4(sp),d1
                move.b  d1,(a3)

***ユーザーモードに戻る
                move.l  spbuf,-(sp)
                dc.w    _SUPER
                addq.l  #4,sp

                rts

*
*スタックバッファ
*
spbuf           ds.l    1

                end
```

**リスト2**

```
10 /* save"f:¥basic¥radicon.bas
20 str z
30 dim int ch(3)={4,4,4,4}
40 initialize()
50 /*
60 while z<>"q"
70 /*
80 z=inkey$
90 switch z
100   case "i":ch(0)=up(ch(0))   :clock(0,ch(0)):break
110   case "m":ch(0)=down(ch(0)):clock(0,ch(0)):break
120   case "k":ch(1)=up(ch(1))   :clock(1,ch(1)):break
130   case "j":ch(1)=down(ch(1)):clock(1,ch(1)):break
140   case "e":ch(2)=up(ch(2))   :clock(2,ch(2)):break
150   case "x":ch(2)=down(ch(2)):clock(2,ch(2)):break
160   case "d":ch(3)=up(ch(3))   :clock(3,ch(3)):break
170   case "s":ch(3)=down(ch(3)):clock(3,ch(3)):break
180   case "q":break
190 endswitch
200 for iii=0 to 3
210 print ch(iii);
220 next
230 print
250 endwhile
260 end
270 /*
280 /*
290 /*
300 func clock(chnum;int,ps;int)
310   prtout(ps*8+chnum*2+1)
320   prtout(ps*8+chnum*2+0)
330   prtout(ps*8+chnum*2+1)
350 endfunc
360 /*
370 /*
380 /*
390 func initialize()
400   int iii
410   for iii=0 to 3
420     clock(iii,4)
430   next
440 endfunc
450 /*
460 /*
470 /*
480 func int up(ps;int)
490   ps=ps+1
500   if ps>7 then ps=7
510   return(ps)
520 endfunc
530 /*
540 /*
550 /*
560 func int down(ps;int)
570   ps=ps-1
580   if ps<1 then ps=1
590   return(ps)
600 endfunc
```

不定期連載ワンチップIC工作(第1回)

# エコーを作る

Takao Katsuhiko 高尾 克彦

**カラオケの音声を際立たせるエコー。声に厚みを持たせる処理はたったひとつのICで実現できるのです。そんな利口者のICを使ったハード工作講座の連載を始めましょう。**

最近のLSIの集積度はどんどん上がっています。ガリウムヒ素がどうしたとか,酸化ケイ素がどーたらこーたら,アルミレールの幅が……とか毎月のように集積回路をめぐる新技術が開発され,どんどん集積度が上がっていきます。

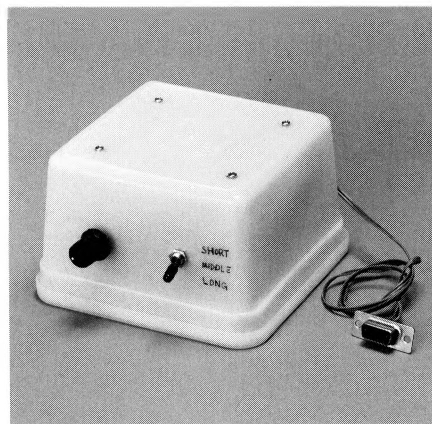
以前なら基板1枚分に相当したような回路が,このような技術を使うとワンチップのLSIになってしまう,というようなことがあちらこちらで起こっています。

最近では,ある機能を実現しようと思ったら,それ専用のLSIを見つけてくれば,それでこと足りてしまう場合が往々にしてあります。いわゆるASSP (Application Specific Standard Products) というやつです。特定の用途向けの回路をぎっしり(たいていは)ひとつのLSIの中に詰め込んでしまったものです。

そのようなLSIを使えば,OPアンプをガシガシつなげたり,難しくて複雑な計算を行わなくても,LSIのデータシートに従っていけばなんとかなります。

そして,読者に代わって,何か面白そうなLSIを見つけ出し,そのデータシートを入手し,それに従って,なんとか回路にしていくのがこの連載です。

データシートには,たいていサンプル回路が掲載されていて,ほとんどはそれらに従って回路を組んでいけば,うまく動作するはずなのですが,なかにはいろいろと調整が必要だったり,少し手を加えないと実用にならないようなものもいくつか見受けられます。このようなサンプル回路はこちらで変更して掲載しますが,原則的にはデータシートに載っている回路をそのまま製作することにします。ASSPを使って,サンプル回路程度ということで,技術的には三沢氏の「ハードウェア工作入門」を理解できている読者が対象になるのではないかと思います。



エコー完成品

## エコー装置を作る

さて,クイズ。

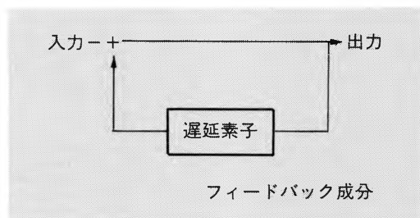
フランク・シナトラ,ポール・アンカ,サリナ・ジョーンズ,エルビス・プレスリー,サミー・デイビスJr.,トム・ジョーンズ。

彼らの共通点はいったいなんでしょう?
答)「And now the ……」のマイウェイを歌った人たちです。

てなわけで,忘年会/新年会のシーズンです。忘年会といえばカラオケで,新年会といえばカラオケです。私はあまりカラオケは好きではないのですが,いまはどこへ行ってもカラオケがあり,よっぽどの田舎でない限り駅前には必ずカラオケボックスがあります。あー,やな世の中になったもんです。

カラオケといえば,あのききすぎるくらいにきいているエコーですね。手始めとして今回はそのエコーを作ってみます。

図1 エコー装置のブロック図




エコー装置というのは,「やっほー」と山に向かって叫べば,しばらくあとに「やっほー」と返ってくるアレを人工的に再現しようというもので,ブロック図で書けば図1になります。

このブロック図の意味するところは「出力信号の一部をもらってきて,少しあとに入力信号に加えてやる」ということです。この図を見て「なんでぃ,遅延素子ってのは,EXP (-jωLt) じゃねえか。おっと,フィードバック成分をそのまま返したら,発振しちまうじゃねえかっ! この馬鹿者!」と思ったそこのあなた,この連載では読者,ライターを問わず,どなたからの飛び入り投稿もお持ちしております。

それはともかく,エコー装置を作るということは,実は遅延素子を探してくればいいということだとわかりました。原始的には地球を7周半回せるだけのコードを買ってきて,それに信号を通せば約1秒のエコーがかけられるわけです。親がKDDの社長でもやってない限り,この方法は使えそうにありませんけどね。

次。信号の流れを川の流れにたとえると,川の流れを遅らせようと思ったら,ダムを建設します。で,水を流したいときに水門を開いて,水をためたいときに水門を閉じるのです。このようなダムを川に沿って作っていけば,放っておくよりもゆっくり水は流れます。ダムの貯水池(こういういい方をするのかどうかわかりませんが,とにかく水のたまっているところです)をコンデンサに,水門をスイッチに置き換えたのがBBD (Bucket Brigade Device) と呼ばれる遅延素子で,アナログ式のエコー装置はたいていこの方法を使っています。

もちろん,ダムの数すなわちコンデンサの数によって分解能が決まってしまうわけで,音質を追求していったり,遅延時間を

増やそうとしたりすると，これらの個数を増やさなければいけないわけで，価格のほうも急激に上がっていってしまいます。

そして第3の方法。X68000のユーザーなら，音というのはPCM方式で記録可能だということは知っていると思います。そして，その逆を行えば元の音が再生できます。これを高速に行えばいいのです。これなら，基本的にはデジタル回路ですし，面倒な調整もあまりありません。再現性もアナログ方式よりはよいでしょう。メモリの価格もかなり下がってきました。

ここで,「そんなこといったって，どうせDRAMを制御しなきゃならないんでしょ。A/Dコンバータも作らなきゃいけないし，その反対のD/Aコンバータも作らなきゃいけないんでしょ。いいの，私なんか。みんなで幸せになってくれれば，それでいいの」などと思ってはいけません。これらの動作をすべて1チップで行うLSIが出ているのです。

ジャジャジャーン。ぱん。

## M50195というLSI

いろいろとカタログを調べてみると，エコー装置の製作にはM50195(三菱電機)というLSIがよさそうです。

さて，図2を見てください。先ほどいった回路がひと通り入っているのがわかります。 ADMというのは，ぶっちゃけていうとAD PCMの一種です。よって，ADM変調器といったらA/Dコンバータのことで、ADM復調器といったらD/Aコンバータのことです。詳しくは囲みを見てください。

これだけの回路が秋葉原で1500円で売っているのです。1500円出せば，地球を7周半できるだけのコードは買わなくてもよくなりました。パチパチ。

## 回路を組む

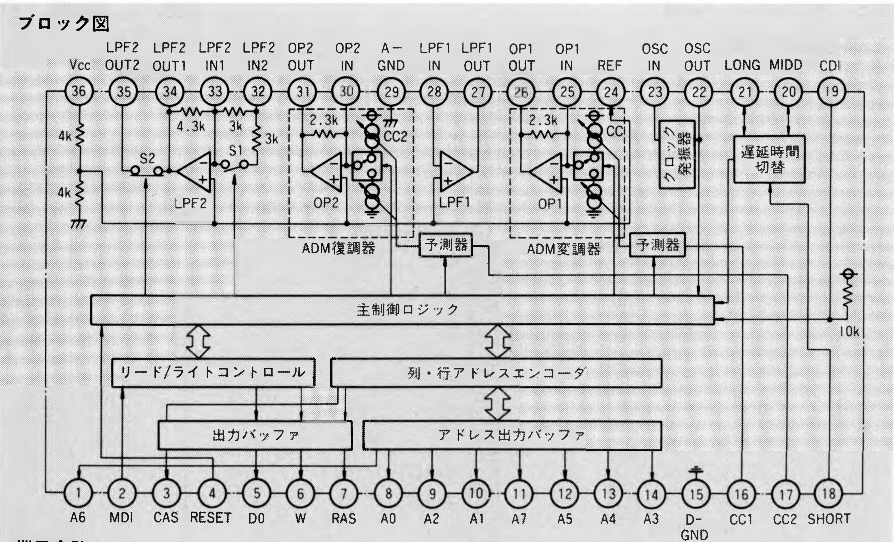
M50195について説明しましたが,あくまでも遅延要素(専門用語で「無駄時間要素」ともいう)についての説明でした。いま作ろうとしているのは，エコー装置なのです。さて図1を思い出してください。これを作っている途中だったのです。M50195だけではエコー装置になりません。

そしてなにより，X68000からのオーディオ出力はステレオ信号で入力は2つ，出力も2ついることになります。そこらへんを考えて設計したブロック図が図4です。これを回路化すると図5-1，5-2になります。

図5-1のほうは,ほとんどデータシートに載っていた回路そのままです。ここらへんについてはあまり質問をしないでください。私にだってよくわかりません。このように組め，とデータシートに書いてあったのです。

図5-2のほうはM50195をステレオで使うための回路です。左下のオペアンプ以外は,加算回路を構成しています。オペアンプを使った加算回路がどうしてあのようになるのかは,103ページのカコミ記事に解説しておきましたので参照してください。

**図2 M50195の内部ブロック図**




**端子名称**

| 端子番号 | 名　称 | 略　称 | 使　用　法 | 標準出力電圧 |
|---|---|---|---|---|
| ① | アドレス6出力 | A6 | メモリのA6(Address Input6) に接続 | 5Vp-o |
| ② | メモリデータ入力 | MDI | メモリのQ (Data Output) に接続 | — |
| ③ | 列アドレスストローブ出力 | CAS | メモリのCAS (Column Address Strobe Input) に接続 | 5Vp-o |
| ④ | リセット入力 | RESET | Lレベルでリセット,電源投入時は外付C,Rでリセットをかける | — |
| ⑤ | データ出力 | DO | メモリのD (Data Input) に接続 | 5 Vp-o |
| ⑥ | ライトコントロール出力 | W | メモリのW (Write control Input) に接続 | |
| ⑦ | 行アドレスストローブ出力 | RAS | メモリのRAS (Row Address Strobe Input) に接続 | |
| ⑧ | アドレス0出力 | A0 | メモリのA0に接続 | |
| ⑨ | アドレス2出力 | A2 | メモリのA2に接続 | |
| ⑩ | アドレス1出力 | A1 | メモリのA1に接続 | |
| ⑪ | アドレス7出力 | A7 | メモリのA7に接続 | |
| ⑫ | アドレス5出力 | A5 | メモリのA5に接続 | |
| ⑬ | アドレス4出力 | A4 | メモリのA4に接続 | |
| ⑭ | アドレス3出力 | A3 | メモリのA3に接続 | |
| ⑮ | ディジタルGND | D-GND | アナログGNDとは一点アースする | 0V |
| ⑯ | 電流制御1 | CC1 | | 0.7V (無信号時) |
| ⑰ | 電流制御2 | CC2 | | |
| ⑱ | ショート | SHORT | Hレベルで遅延時間Td=100msec, インジケータ用出力電流5mA標準 | 5V (S時) 0V (M, L時) |
| ⑲ | コンパレータデータ入力 | CDI | コンパレータの出力に接続 | — |
| ⑳ | ミドル | MIDD | HレベルでTd=150msec, インジケータ用出力電流5mA標準 | 5V (M時) 0V (S, L時) |
| ㉑ | ロング | LONG | HレベルでTd=200msec, インジケータ用出力電流5mA標準 | 5V (L時) 0V (S, M時) |
| ㉒ | 発振器出力 | OSC OUT | 4MHzセラミックフィルタを接続, 外部クロック使用時はオープン | 5Vp-o |
| ㉓ | 発振器入力 | OSC IN | 4MHzセラミックフィルタを接続または外部クロックを入力 | — |
| ㉔ | リファレンス | REF | ≒1/2Vcc | 2.5V |
| ㉕ | オペアンプ1入力 | OP1IN | 外付Cによりミラー積分器を構成 | 2.5V |
| ㉖ | オペアンプ1出力 | OP1OUT | | 2.5V |
| ㉗ | ローパスフィルタ1出力 | LPF1OUT | 外付C, Rにより2次ローパスフィルタを構成 | 2.5V |
| ㉘ | ローパスフィルタ1入力 | LPF 1 IN | | 2.5V |
| ㉙ | アナログGND | A-GND | | 0V |
| ㉚ | オペアンプ2入力 | OP2IN | 外付Cによりミラー積分器を構成 | 2.5V |
| ㉛ | オペアンプ2出力 | OP2OUT | | 2.5V |
| ㉜ | ローパスフィルタ2入力2 | LPF2IN2 | 端子㉛, 端子㉜間外付Cによりローカットフィルタを構成 fc 2 =240Hz標準(SHORT時), fc' 2 =120Hz標準(MIDD, LONG時), 端子㉝, 端子㉞間外付Cにより1次ローパスフィルタを構成fc 3 =5.7kHz標準 | 2.5V |
| ㉝ | ローパスフィルタ2入力1 | LPF 2 IN 1 | | 2.5V |
| ㉞ | ローパスフィルタ2出力1 | LPF2OUT1 | | 2.5V |
| ㉟ | ローパスフィルタ2出力2 | LPF2OUT2 | 帰還抵抗を接続し入力へ帰還させる | 2.5V |
| ㊱ | 電源 | Vcc | 4～5.5Vを印加 (定格5V) | — |

まず，最初の（左上の）加算回路ではステレオ信号の右（R）と左（L）を足し合わせています。RとLそれぞれに，エコー回路を組んでもいいのですが，回路が2倍になってしまいますし，人間の耳はそれに見合うだけの能力を持っていないようなので，エコー成分の作成にはRとLを足し合わせた信号を用いることにしました。

このままモノラル信号を出力してもいいのですが，せっかくX68000がステレオ出力していることですし，分離してやりましょう。エコー成分を元のRとLに足し合わせてやっているのが，2つの右側のオペアンプです。

さて，左下のオペアンプが残っていました。これはいったいなにをするのでしょう。

このエコー装置の電源はX68000のジョイスティック端子から取っていました。本当はこの電源はノイズが多いのでオーディオ関係に利用したくないのですが，簡略化のために今回は採用しました。この連載の

**図3　M50195のピン配置**

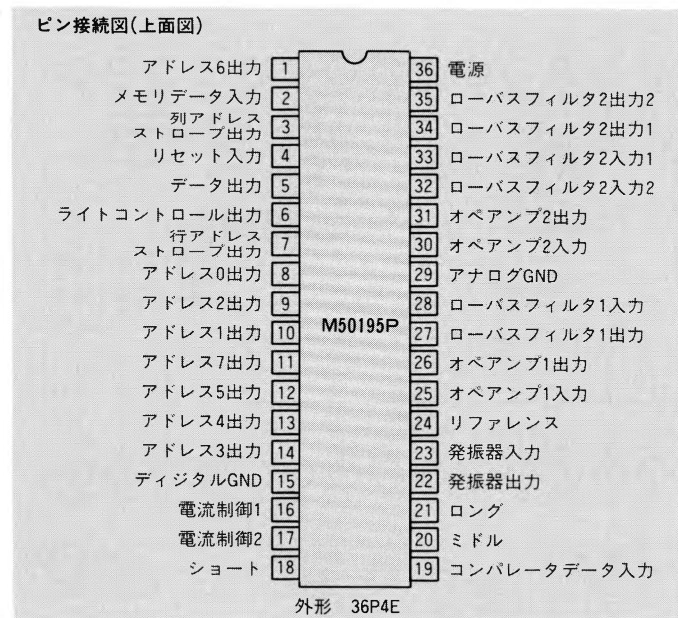



**図4　エコー装置のブロック図**

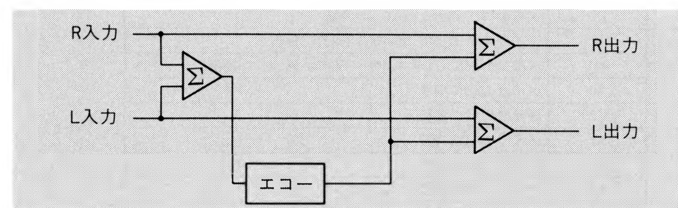



**図5-2　回路図（その2）オペアンプ周り**

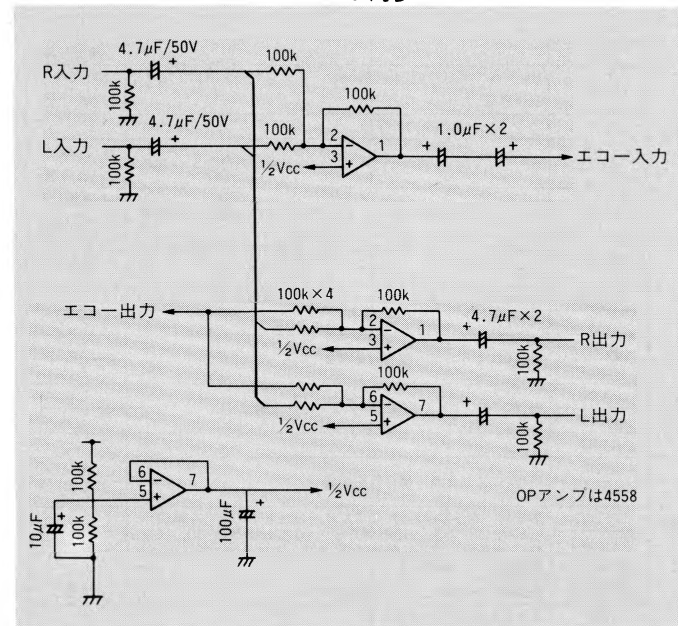
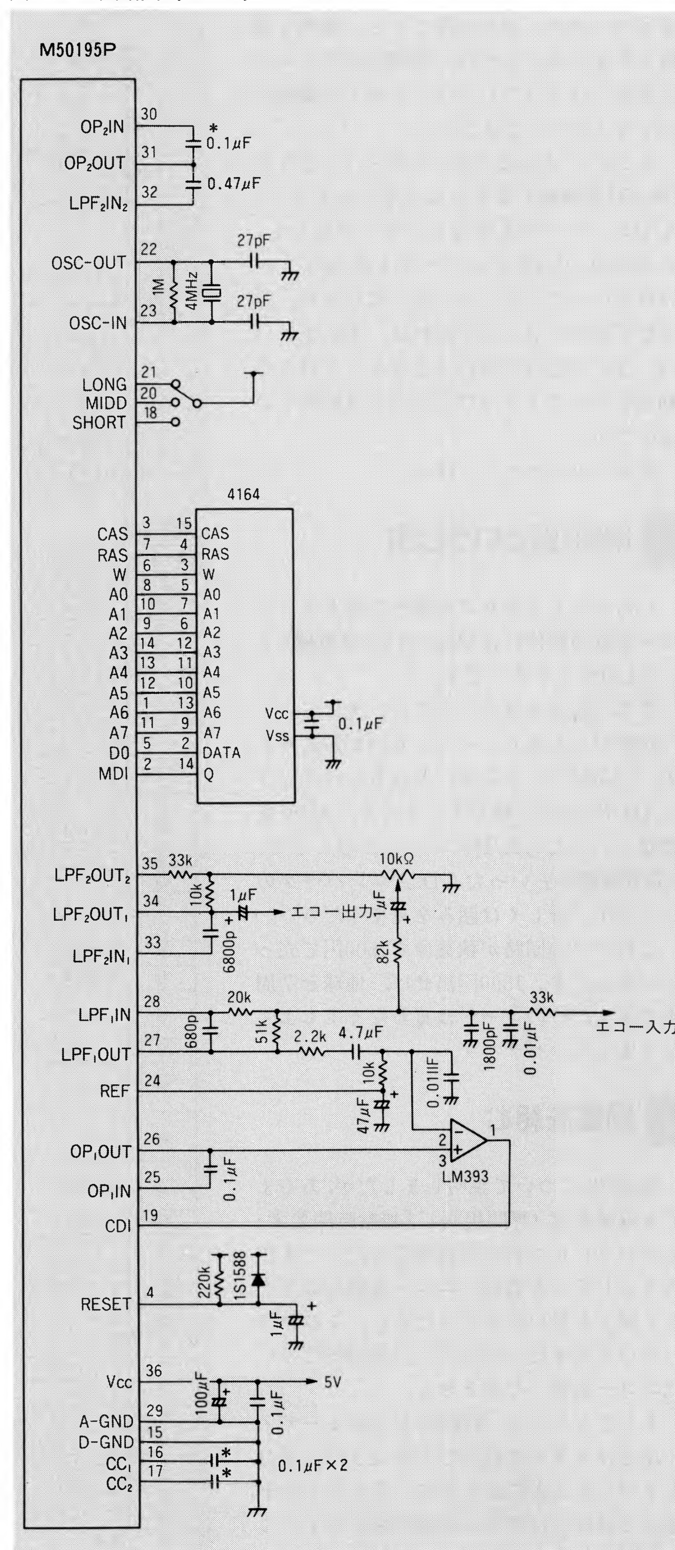



**図5-1　回路図（その1）M50195周り**



回が進んで，ひと通りのエフェクタを作り終えたら，まとめて電源を作成しようと思います。それまで待てない人は，1992年11月号の「ハードウェア工作入門」に出ていた電源回路を作成してください。アナログ回路でもデジタル回路でも，電源は同じですので共用できるはずです。

話を元に戻すと，電源の話をしていたのでした。普通のオペアンプは－12V，0V，＋12Vの電源を必要とするのですが，それだけの電源を用意するのは大変なので，今回は音質を犠牲にして（実際に聴き比べたわけではありませんが，多分犠牲になっています），0Vと＋5Vという片電源で動作させています。つまり，

－12V　→　0V

＋12V　→　＋5V

となります。そうすると，信号の基準点となる0Vは，

0V　→　＋2.5V

となるので，その基準信号を作っているのが左下のオペアンプなのです。

音質のためなら電源を作る手間はいとわないという人は，この左下のオペアンプはいりません。そして，1/2Vccにつながっている信号はすべてグランドに落としてください。

## 部品など

まず，M50195のピン間隔が普通のICとは違っています。これはシュリンクパッケージとかMILパッケージとか呼ばれるもので，普通のICはピンとピンの間が2.54mmなのに対して，その2/3になっています。残りのICなどはピン間隔が2.54mmなので，どうにかして，M50195の信号を2.54mm間隔に引き出してやらなければなりません。私はこのために，サンハヤトのMIL ICB-01という変換基板を使用しました。これはシュリンクパッケージの40ピン分を2.54mm間隔に引き伸ばすためのものです。まわりに少しばかり蛇の目基板がくっついているので今回はそこに残りの部品を配置しました。

「ハードウェア工作入門」と違って，今回の製作は一気に部品点数が増えています。図7を見て，これは手ごわいなと思った方は，もうひとまわり大きな基板を使ったほうがよさそうです。確か，64ピン用の変換基板がサンハヤトから発売されているはずです。

それから，M50195のソケットもシュリンクパッケージ用のものを用意しなければなりません。ピンの間隔がずれていてはソケットの役割を果たしません。私は36ピン用のものを見つけられなかったので，40ピン用のものを買ってきて，残りの4ピンを余らせて使いましたが，できるなら36ピン用のソケットを用いたほうが見栄えがよくなるでしょう。

どうも，40ピンと36ピンではICの幅（縦ではなくて横の幅）が違うようで，40ピン用のソケットに強引にM50195を押し込むには，ラジオペンチで少しピンを広げてやらなければなりませんでした。写真からもわかるように，実はソケットに一度はめ込んでしまうと，いわれなければわからない程度になってしまうんですけどね。

M50195のほかに，LM393とRC4558というICがあります。これはオペアンプと呼ばれるICで，名前のアルファベットの部分が製造メーカーを表し，数字の部分が型番を

**図6　LM4558，LM324のピン配置**

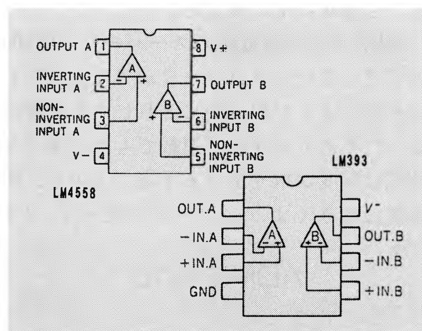



**図7　実体配線図（表）**

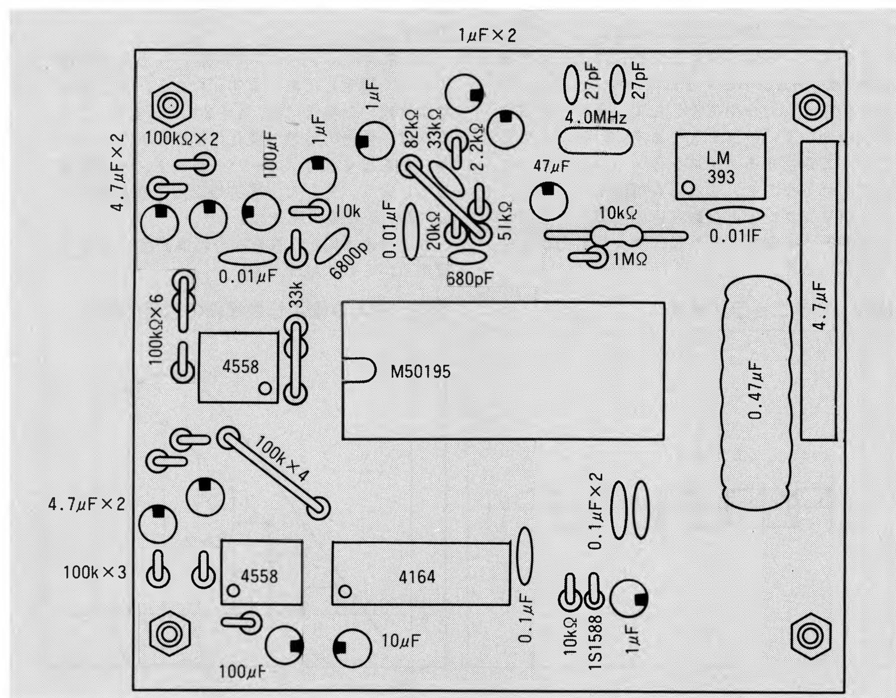
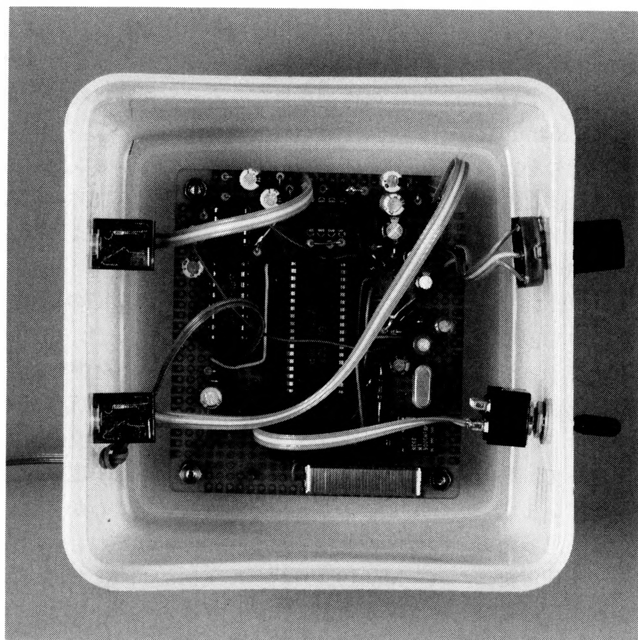



**表1　部品表**

| | 数量 | 単価 |
|---|---|---|
| LSI | | |
| M50195 | 1 | 1500 |
| M4164 | 1 | 480 |
| オペアンプ | | |
| RC4558 | 2 | 80 |
| LM393 | 1 | 80 |
| ダイオード | | |
| 1S1588 | 1 | 20 |
| 水晶発振機 | | |
| 4MHz | 1 | 300 |
| 抵抗 | | |
| 2.2kΩ | 1 | 10 |
| 10kΩ | 2 | |
| 20kΩ | 1 | |
| 33kΩ | 2 | |
| 51kΩ | 1 | |
| 82kΩ | 1 | |
| 100kΩ | 15 | |
| 220kΩ | 1 | |
| 1MΩ | 1 | |
| コンデンサ | | |
| 27pF | 2 | いろいろ |
| 680pF | 1 | |
| 6800pF | 1 | |
| 0.01μF | 3 | |
| 0.1μF | 4 | |
| 0.47μF | 1 | |
| 4.7μF | 1 | |
| 電解コンデンサ | | |
| 1μF | 5 | いろいろ |
| 4.7μF | 4 | |
| 10μF | 1 | |
| 47μF | 1 | |
| 100μF | 2 | |
| ボリューム | | |
| 10kΩ | 1 | いろいろ |
| スイッチ | 1 | いろいろ |
| 基板 | | |
| MIL ICB-01（サンハヤト） | 1 | 670 |
| ステレオコネクタ | 2 | 120 |
| 9ピン　D-SUBコネクタ | 1 | 200 |

合計で，だいたい5千円くらい



配線のようす

表しています。よって，部品表にあるものとまったく同じである必要はありません。たとえば，LM393の代わりにMC393でもかまいませんし，RC4558の代わりにμA4558でもかまいません。

回路図の中に4.7μFとか0.47μFとかという値を間違えたんじゃないかと疑いたくなるような大きな容量のコンデンサが見えますが，間違いではありません。私はなんとかしてこれらを見つけましたが（写真の右奥のほうのデカい塊2つがそれ），これらの代わりに無極性の電解コンデンサでもかまいません。これも見つからなければ，図8に示すように同じ容量の電解コンデンサを2つ使って代用してください。

残りのコンデンサは，なんの変哲もない普通のものを使っていいのですが，回路図中＊印を付けたところのコンデンサは精度が直接音質にかかわってきますので，もし気になるようでしたら，精度のよいものにしてください（当然，値段も高くなります）。データシートには「高精度フィルムコンデンサ（±5％以内）」を使用のこと，と書かれていました。

今回，ケースは近所のスーパーに3個100円で売っていたタッパーウェアのようなものを使いましたが，別にこれにする必要はまったくありません。連載でもエフェクタ関係は何度か取り上げる予定ですので，積み重ねのきくようなケースを使って，（ミニ）コンソールボックスを構築できるようなものを選んでおいたほうがよいかもしれません。

## 製作

部品が揃ったら，あとはただひたすらハンダ付けをして，回路を組み立てるだけです。ハンダ付けの仕方や，道具の使い方やらはいいですよね。

で，組み上がったら，まだM50195を差さずに，それ以外のICをすべて差して電源を入れます。変な臭いがしたり，どこか急激に熱くなったりしていませんか？　古典的な質問ですが，最近のICはけっこう丈夫にできているのでこれさえクリアすれば，まぁ大丈夫でしょう。

次に，図5-2で左下にあるLM4558の7番ピンにテスターを当てて，電圧を測ります。前述した基準電圧2.5Vがうまく生成されているかどうかをチェックします。うまく生成されているようなら，プラグ類を接続し，音楽をかけてみます。ちゃんと聞こえますか。音量は半分くらいになるかもしれませんが，聞こえていればOKです。M50195をまだ差していないのに，エコーがかかっていたりしませんか？

以上のチェックがすべて完了したら，図5-2に示されている部分はうまく作動して

### ADMについて

今回作成したエコー装置は一度音をデジタル信号に変換してRAMにバッファリングしておき，それを再生してエコー効果を出すものでした。

アナログ信号をデジタルデータに変換するのはよいのですが，その変換したデータを収めるRAMが4164ひとつしか見当たりません。この4164というのは，1ビット×64Kビットタイプのdramです。つまり，今回のシステムにはRAMは1ビット幅分しかないのです。

ということは，出力される信号はある/ない式の音？　そんなことはありません。そのからくりがADM(Adaptive Delta Modulation：適応差分変調)なのです。

まず，ADMの基本となるDM(Delta Modulation：差分変調)について説明しましょう。

音の波形というのは，人為的に（悪意を持って）操作しない限り，連続です。この場合，連続というのはグラフ化したときにグラフが途切れないということです。つまり，今の音量に対して次の音量は増加/変化なし/減少と判別できるのです。とりあえず，変化なしというのは置いておき，

増加　→　1
減少　→　0

と対応させておきます。そして変化なしというのは，101010……で表します(図9)。このようにすれば，音を1ビットで表現できるのです。これがDMです。

やはり，音量の変化がないのに，101010……で表していたのでは音になってしまいます。このようにして出てしまったノイズをグラニュラノイズ（粒子性雑音）といいます。

このノイズを低く抑えるためには，1回の増加分/減少分Δを小さくすればいいのですが，これを小さくしたら小さくしたで，今度は急激な変化についていけなくなります(図10)。このようにして発生するノイズをオーバーロードノイズというそうです。これらを解決するために，サンプリングレイト（取り込み間隔）を上げたりすると，必要なメモリ量が莫大なものになってしまいます。

ここで新しい仮説を入れます。それは，「音量が連続して増えるときには急激に増えるし，連続して減少するときには急激に減る」というものです。

Δをグラニュラノイズが問題にならない程度に小さく設定しておきます。ただし，同じ符号が二度続いた場合には，Δ＝Δ×2します。こうすれば，信号の急激な変化に追いつかないということもなくなり，オーバーロードノイズも減るでしょう。そして，異符号が出てきた時点で，このΔは元の値に戻されます。

以上のようにΔの値を符号の連続度により可変する。これがADMです。

図9　グラニュラノイズ

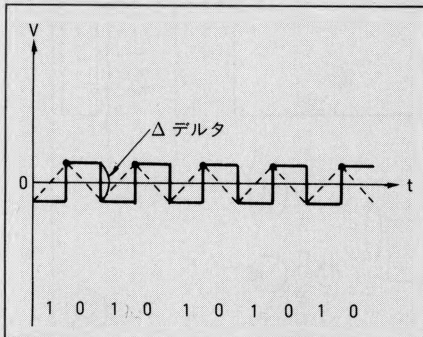
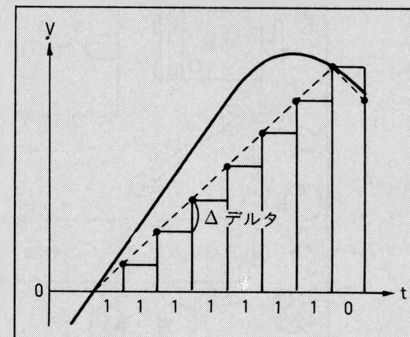



図10　デルタ幅より急傾斜の信号の場合



います。これから先にトラブルが起こっても図5-1の部分を調べればよいのです。逆に，これまでのチェックにどこかしら引っ掛かったら，図5-2の部分を厳重に調べてください。

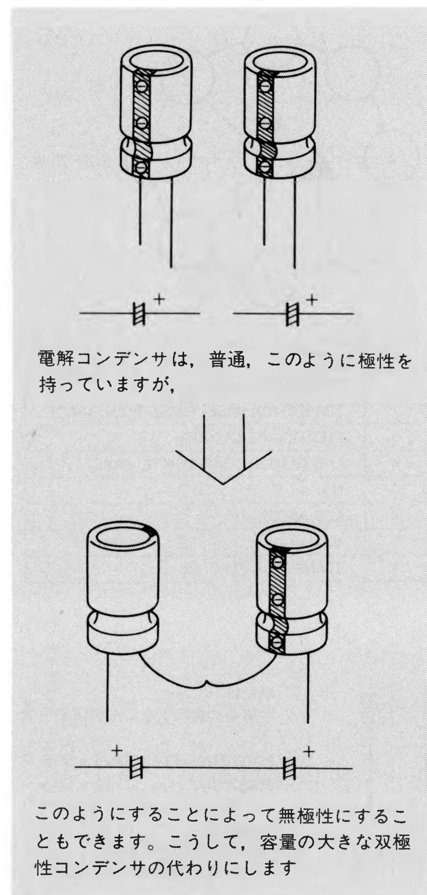
次に，M50195のリセット端子(4番ピン)を調べてみます。M50195のリセット端子は，Lレベルで有効なので，テスターで測ってみて，ちゃんと5Vになっているかどうかを確かめます。ここが0Vになっていると，残りの回路はせっかく正常に作動しても，M50195にリセットが掛かりっぱなしになってしまうので，うまくエコーがかかりません。

以上のチェックがすんだら，いよいよエコー機能のチェックに入ります。一度，電源を抜いてM50195をソケットに差し込み，もう一度電源を入れます。ディップスイッチのいずれかをONにしてください。スピーカー（またはイヤホン）からエコーがかかった音声が出力されていますか？　このときに，エコーのききまくった演歌をかけてはいけません。どちらのエコーが作動しているのか判定できなくなります。

エコーがうまくかかったら，スイッチをいろいろ変えてみてください。エコーのかかり具合が変わりますか？　このスイッチはエコーの遅延時間の調整です。いろいろやってみて，好きなところを選んでください。ボリュームは，エコーのフィードバック量の調整です。戻ってくるやまびこの大きさを調節します。これも好みのところに合わせてください。どちらに回しすぎても，発信したり，振り切れたりしません。安心して調整してください。

で，基板をケースに固定し，蓋を閉じたら完成です。

**図8　電解コンデンサを無極性にする**




電解コンデンサは，普通，このように極性を持っていますが，

このようにすることによって無極性にすることもできます。こうして，容量の大きな双極性コンデンサの代わりにします




## 次回(来月とは限らない)の予定

メーカー発行のデータシートに載っているサンプル回路をいろいろ紹介していこうというこの企画，いかがだったでしょうか。

次回は（本当はカラオケは嫌いなんですけど），プリアンプとミキサーを作成して，このエコー装置にマイクを取り付けられるようにしよう，とかエフェクタシリーズとかいってトレモロとかコーラスなどを作ろうか，などと思っています。まぁ，あせらずに1つひとつ作っていきましょう。

また，パソコン誌にハード関係の記事が極端に少なくなった今日この頃，読者の皆さんが，どのくらいのレベルからを難しい，簡単だ，と判断するのかいまひとつ把握できません。「こんなの，いちいち記事にするなよー」とか「今回のはよくわからない。もっと丁寧に説明しろ」とかでもいいですから，読者の皆さんの反響が知りたいと思っています(もちろん，「こんな連載とっとと終わらせろ」でも悲しいけどありがたい)。

それとは別に，「こんなものがほしい」というリクエスト（←実現できるかどうかはわかりませんが)や「こんなLSIがあるんですけど」というタレ込みやらも大歓迎します。文中でもいったように，私の代わりに記事を書いてきてくれても結構です。分量やスタイルは，今回の記事をひとつの標準としましょう。

それでは，また次回まで。73&88&バイバ～イ。

**参考文献**
1)　三菱電機，M50195データシート
2)　松村　南，デジタル・エコー装置の製作，トランジスタ技術，1990年5月号
3)　富沢　瑞夫，ディジタル遅延用ICの使い方，トランジスタ技術，1987年11月号

### オペアンプの加算回路について

三沢氏の「ハードウェア工作入門」のほうで以前にやったことがあるとばかり思っていたのですが，探してもなかなか見当たらないので，ここでオペアンプによる加算回路の仕組みを説明します。

まず，図を見てください。これが加算回路の基本です。ここで，端子1，端子2からそれぞれ$I_1$，$I_2$Aの電流が流れ込んできます。オペアンプに入る電流は無視できるくらい少ないですから，$I_1$，$I_2$は，ほとんど$R_3$を伝わって出力に流れていきます(「入力インピーダンスが無限大である」という)。

いま，図の矢印をつけたところを基準点にすると，これらの関係は

$$V_1 = R_1 * I_1$$
$$V_2 = R_2 * I_2$$
$$V_{out} = R_3 * (I_1 + I_2)$$

となりますので，入力電圧$V_1$，$V_2$と出力電圧$V_{out}$の関係は，

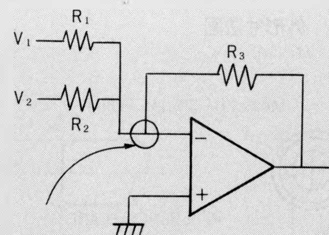
$$V_{out} = R_3 \left( \frac{V_1}{R_1} + \frac{V_2}{R_2} \right)$$

となります。

ここで，$R_1 = R_2 = R_3$ならば，

$$V_{out} = V_1 + V_2$$

となります。つまり，これは足し算を行っているわけで，図の回路は加算回路として機能するわけです。



X68000にバーコードリーダーをつなぐ

# バーコードリーダー作るんですか?

Ishigami Tatsuya 石上　達也

**さまざまな分野で使用されるバーコード。ここではX68000にバーコードリーダーを接続してみましょう。読み取り率は?ですが，かんばればバーコードバトラーもできるかな?**

バーコード。この言葉を知らない人はまずいないでしょう。商品やその包装紙についている，白地に縦の黒棒の入ったアレです。Oh!Xでも昨年の2月号から裏表紙に，このバーコードが印刷されるようになりました。

バーコードのついた商品をお店で買おうとすると，レジでこの部分を機械に読み込ませ，いくらの商品であるかを計算します。レシートに出力されるのは商品の値段だけですが，お店の中ではこれらのデータをストックしておき，いつ，どのような形態で(まとめ買いか否かなど)商品が売られたのかを分析し，マーケティング展開の指針とするようです。

以前，3段ブチ抜きで「情報は武器だ」なんて週刊誌の広告がありました。現在，なにが売れているのか，なにを出せば売れるのか，というマーケティング情報は技術力にも劣らないくらい重要な武器のようです。

そして，これらのバーコードを読み取る装置が今回製作するバーコードリーダーです。

## バーコードリーダーの種類

さて，今回はこのバーコードリーダーを作るのですが，いきなりオリジナルなものを作るのは大変なので，現在市場に出回っているバーコードリーダーをちょっと調べてみましょう(分類は参考文献1による)。

○ペン式

光源として発光ダイオード，受光素子としてフォトダイオードが，ペン状の筐体に収められている。発光ダイオードを出た光はバーコード面に反射し，フォトダイオードに吸収される。フォトダイオードはこの光を電気信号に変換し，この電気信号からバーコード面の反射率を逆算し白/黒の判定を行う。光の反射率で判定を行うので，バーコード面がビニール製であったり，曲面であったりすると不利。また，ペンを人手で移動させるので，ペン速度が一定になるように多少の熟練が必要となる。もしくは，読み取り精度が悪い。

○タッチ式

ずばり，イメージスキャナでバーコードを読み取ってしまう方式(誰かHAL研のハンディスキャナでバーコード読み取りプログラムを作ってみませんか?)。画像そのもので判定を行うので，理論的には人間がバーコードと認められれば読み取りが可能。イメージスキャナが必要ということで高価。

○レーザー式

ハンディタイプと定置型がある。基本的には，発光ダイオードの代わりにHe-Neレーザーを使い反射率を調べる構造。レーザープリンタのノリでレーザーをあらゆる方向に飛ばし，反射して戻ってきたところにバーコードがあるものとして，信号を読み取る。機械的な可動部が多く価格が高いため，比較的大きなスーパーマーケットやコンビニエンスストアなどに設置されている。

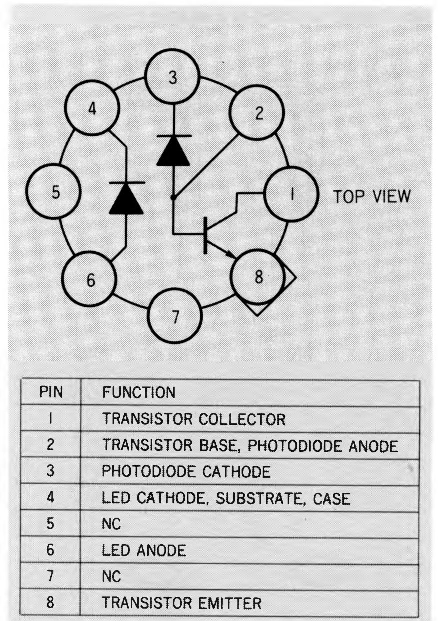
＊　＊　＊

今回は，安く作るということで，ペン方式のバーコードリーダーを製作することにします。

## HBCS-1100という部品

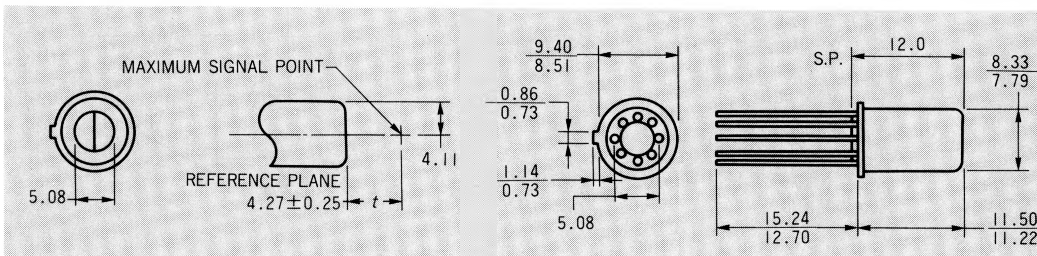
参考文献4からもわかるように，標準バーコードの最細線は0.330mmです。ということは，発光ダイオードからの光は少なくとも直径0.330mm以下のスポットにしな

図1.1　HBCS-1100 内部接続図




| PIN | FUNCTION |
|---|---|
| 1 | TRANSISTOR COLLECTOR |
| 2 | TRANSISTOR BASE, PHOTODIODE ANODE |
| 3 | PHOTODIODE CATHODE |
| 4 | LED CATHODE, SUBSTRATE, CASE |
| 5 | NC |
| 6 | LED ANODE |
| 7 | NC |
| 8 | TRANSISTOR EMITTER |

図1.2　外形寸法図



ければなりません。このような細かい精度の工作は，自作するより完成品を買ってきたほうが利口というものです。今回は横河ヒューレット・パッカード社のHBCS-1100という部品を使用することにします。このHBCS-1100は，

○発光部・受光部を単一パッケージに内蔵
○高分解能。0.19mmのスポットサイズ（パッケージ前面4.27mmにおいて）
○700nmの可視光を使用
○外部からの光をレンズフィルタでカット
○TO-5小型シールドパッケージ
○PINフォトダイオード，トランジスタ出力
○高信頼性

などの特徴を持つ（参考文献3より），高分解能オプティカルリフレクティブセンサです（なんて，バイリンガルでアメリカンな言葉だ）。自分で高信頼性とか高分解能などと書いていると，高級紳士靴下とかいって3足1,000円で売られているカラーソックスみたいで，ちょっぴり恥かしいものがありますが，とにかくこれを利用するのが賢明そうです。

私がHBCS-1100を秋葉原の光南電子（ラジオデパート2階）で買ったときの値段が6,200円だったので，値段のほうも高級です。今回の電子回路の主役ですが，出費のほうでも主役です。

そんなHBCS-1100のダイアグラム＆内部等価図は図1です。

## 電気回路

図2が今回製作するバーコードリーダーの回路図です。以前のアナログスティックの製作に比べれば，ずいぶんとシンプルなものです。

図1からもわかるように，HBCS-1100は内部に増幅用トランジスタを内蔵していますが，このトランジスタを使って信号の増幅を行うには，線形性や温度特性などの面で若干の不安が残ります。ここではLM358というOPアンプを用いてフォトダイオードからの微弱電流を2～4Vの電圧に変換しています。このLM358というOPアンプは，ハードウェア工作入門でもお馴染みですね。

そんなわけで，今回はHBCS-1100に内蔵のトランジスタは使用しないので，参考文献3にあるように「ディテクタ部IC基板のダイオード特性を打ち消すためにトランジスタのコレクタを＋Vccに接続し，ベースとエミッタをショートさせてください」だそうです。

また，フィードバック抵抗に15MΩというたいへん抵抗値の大きなものを使用していますので発振防止用に，この抵抗とパラレルに0.47μFのコンデンサを入れておきます。

このまま出力信号をADコンバータもしくはコンパレータ（比較器）にかければ反射率測定器の完成なのですが，我々がほし

### 信頼性のほしい人のために

今回の製作は，なるべく安くという目標のもとにすすめたものです。今回の記事を参考に，八百屋さんがレジスタの代わりにX68000を用いて在庫管理システムを作ろうとか，工場の塗装ロボットの制御にX68000を使おうとかいうのはあまりおすすめできません。べつにX68000を実用的に使うなといっているのではなく，今回のバーコードリーダーでは読み取り精度が非常に悪いのです。

もちろん，センサの取り付け部の寸法精度やOPアンプまわりのノイズ対策をちゃんとやれば，市販品と同じくらいの精度が出せるようになるとは思いますが，そこまでやるのは，個人には困難で，そういうときには，多少の出費をしても市販品を買うほうが賢いと思います。

ここでは，そのような方々のために，横河ヒューレット・パッカードのカタログから，X68000に使えそうなものをみつくろってみました。なぜ，シャープでなく，オムロンでもなく，さっきからYHPなのかは，あまり考えないでください。金一封いただいていたり，テレホンカードをいただいたわけではありません。最初にYHPに電話したら，必要なデータ＆カタログをすべて無料で送ってくださったので，他社に資料請求する必要がなくなってしまったのです。そんなわけで，他社の製品は見ていませんのでご了承ください。

今回作ったバーコードリーダーとほぼ同じものをペン型にして，動作スイッチを付けたものがデジタルバーコードワンドHEDS-3000です（もちろん，工作精度は比べものにならないくらい高いです）。これのケースやケーブルにシールドを施し，スイッチを取り除いたものが，同シリーズのHEDS-3050です。どちらも9ピンのD-タイプコネクタが付いていて，一瞬オヤッと思ったのですが，電源ピンが7番ピン（GND），9番ピン（3.6～5.75V）なので直結は無理かと思います。

また，バーコードバトラーのように，隙間にバーコードの書かれた紙を通すタイプの装置（スロットリーダというらしい）としてHBCS-7000シリーズがあります。動作温度や，使用する光の波長，精度などで4種類のタイプがあります。

YHPのカタログには「ホビーから，ビジネス，FAまで」と書いてあったのですが，これらがホビーに当たるのでしょう。YHPのような電機メーカーが，ホビーのことまで考えて製品を作ってくれているとは，嬉しくなってしまいます（本当か？）。

で，ビジネスのほうです。たとえば，威勢がよいついでにリーダーを持つ手にも力が入ってしまい，普通のリーダーでは握り潰してしまうぜい，という八百屋さんや肉屋さんには，金属ケースのHBCS-6000シリーズがおすすめです。「2トンの衝撃にもビクともしない耐衝撃性を有している」そうですので，2トントラックが家にある引っ越し屋さんは試してみるのもいいかもしれません。なお，乱闘に巻き込まれやすい外野席のビールの売り子は，このシリーズ以外を使用すべきでないことはいうまでもありません。

さらに，このシリーズより上のタイプは，読み取り部の密閉にサファイアを用いていて，防塵性も抜群ですから，揚げもの屋さんや海の家のPOSには最適でしょう。また，5mAという非常に小さな電流で動作しますので，バーコードもいいけど，バッテリーがなぁー，という屋台の焼き芋屋さんにもピッタリです。これなら，あの重いバッテリーを担がずに，乾電池1個でこと足ります（パソコン用の電源は知らない）。

家が日焼けサロンでお客の会員カードをバーコードで管理しようという方は，漏れて入ってくる外部光にも注意しなければなりません。でも，HBCS-A000シリーズがあればもうだいじょうぶ。100kLuxまでの外部光の影響をいっさい受けません。また，このシリーズは「新開発のサファイアチップ一体化ポリカーボネードケースの採用により，外部回路を外界から絶縁」しているそうですので，家が熱帯魚ショップで「うちの電気ナマズが放電して困るわ～」という人にもオススメです。15000Vの高電圧においても放電しないことが保証されています。

ハードウェアの問題ではなく，ソフトウェアの問題として，バーコードの規格というものがあります。今回のソフトウェアは，共通商品シンボルのJANというタイプのものですが，図書館の蔵書管理に使われているのは，たいていがNW-7と呼ばれるものですし，産業ロボットのやり取りに使われているのは，たいていはCODE 3 of 9という規格です。

せっかく，X68000に目張りして雰囲気に対する発火性を抑えたのに，CODE 3 of 9読み取りプログラムを作らなきゃいけないのか～，と落ち込んでしまったあなたもHBCR-8000シリーズを使えばもう解決。このシリーズはデコーダ（解読器）を内蔵していて，CODE 3 of 9, Interleaved 2 of 5, UPC, EAN, JAN, NW7, コード128, MSIコード，コード11を自動認識。しかも，その内容をシリアル回線で出力してくれるという親切さ。その出力をマキシム社のMAX232というICで12Vまで昇圧すればRS-232Cに直結できるぞ。さあ，君もX68000で溶接ロボットと友達だ。

※）以上の内容はあくまでも参考程度にとどめておいてください。実際の購入にあたっては，各自でメーカーに問い合わせることをお勧めいたします。

問い合わせ先
横河ヒューレット・パッカード（株）
東京　03-3335-8152
横浜　045-313-1346
大阪　06-300-3147
名古屋　052-571-2270

いのは，バーコードリーダーです。バーコードリーダーで必要なのは，紙の反射率ではなく，紙が白であるか黒であるかという情報です。いくら色が白くても反射率の小さな紙の上では黒と見なされてしまいますし，逆もまた起こりえます。紙の種類を変えるたびに，白黒判定の閾値を設定するのもエレガントな方式ではありません。そこで登場するのが，ショットキーバリアブルダイオードを用いたΔV電圧発生回路です。

LM358で増幅された信号がコンパレータのLM311に入力されるのですが，そのあいだにこのΔV電圧発生回路がぶら下がっています。

コンパレータの入力電圧がピーク値を示すとき（バーコードの白線部分に光が反射したとき）にはその最大電圧－Ｖｆ（電圧降下：実際のダイオードに順方向に電圧をかけても，なかなか抵抗値は０にならない。そのためにダイオード内で下がってしまう電圧のこと）がコンデンサにかかります。コンデンサとコンパレータの出力端子とのあいだには，10MΩと100KΩの２本の抵抗がぶら下がっていて，その中間点にコンパレータの基準電圧が入力されていますね（２番ピン）。つまり，

閾値電圧＝

$$\frac{\text{コンデンサの電圧}\times10^6+\text{出力信号}\times100\times10^3}{10\times10^6+100\times10^3}$$

≒コンデンサの電圧

となりますので，コンパレータの入力電圧がコンデンサの電圧を割ってしまったら，出力は０となります。

逆に，コンパレータの入力電圧が下降する場合を考えてみましょう。電圧はダイオードを通じて，コンデンサを放電させながら下がっていきます。入力電圧が底値にたどりついたときに（バーコードの黒線部分に光が当たったときに）コンデンサの電圧はその最小電圧＋Ｖｆになります。電圧が上昇するときと同様に，この電圧は10MΩと100KΩの２本の抵抗で分圧されて，コンパレータの基準電圧を作っています。例によって，

閾値電圧　＝

$$\frac{\text{コンデンサの電圧}\times10\times10^6+\text{出力信号}\times100\times10^3}{10\times10^6+100\times10^3}$$

≒コンデンサの電圧

ですので，今度はコンパレータの入力電圧がコンデンサの電圧よりも大きくなれば，コンパレータの出力は１となるわけです。

なお，参考文献2ではここのダイオードにヒューレット・パッカード社の5082-2800を使っていましたが，一般にこのパーツは高価かつ入手困難なので，代わりに入手が比較的容易と思われる1SS97を使っています。

## 部品の入手&製作

とりあえず，HBCS-1100以外は特殊な部品は使っていませんので，これさえ入手できれば，地方に在住の方も製作できると思います。この部品の入手がどうしても困難な場合には，横河ヒューレット・パッカードコンポーネント部まで，問い合わせてみてください。

あとは，X68000のジョイスティック端子には９ピンのD-SUBコネクタがつながるって知ってましたよね。表１に部品表を示しますので，これを参考に秋葉原や日本橋を散歩するなり，「トランジスタ技術」の流通広告を片手に通販を利用するのもよいでしょう。

今回ジョイスティック端子から引っ張ってくる信号はVccとGNDと入力信号だけですので，信号線は３本束になったものでよいです。そういうものが見当たらないときは，10芯線を買ってきて，うち３本だけを手で引きちぎって使ってもOKです。

部品が揃えば，回路図どおりに結線すればよいのですが，回路図が読めないよ～という方のために，図３-２に実体配線図を示しておきます。

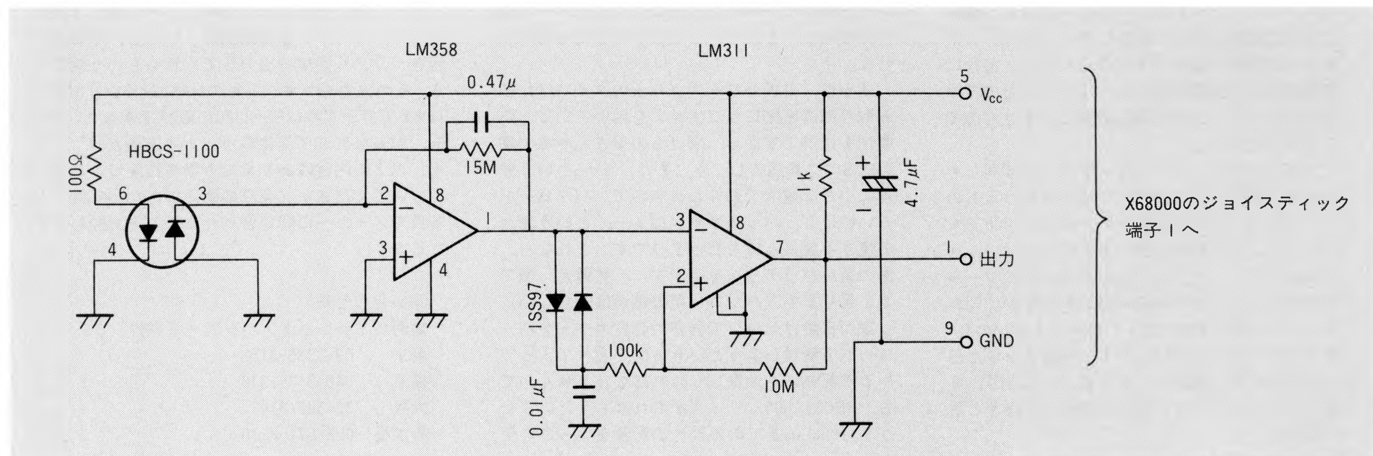
それから，例によって回路図には書いてありませんが，OPアンプの電源ピンのところには，パスコン（容量が0.1～0.5［μF］ぐらいのコンデンサ）を渡しておくとなにかと安心です。また，回路図中，LM358のフィードバック抵抗に15MΩのものを使っていますが，入手できなければ，10MΩ＋4.7MΩで代用してください。

ちなみに，この図版原稿の作成には，COMPACのSimple-CADを使用しました。このソフトは値段のわりに，機能は高く，なぜ注目されないのか不思議でしかたありません。私の個人的希望として，ウィンドウまわりの操作をSX流にするバージョンアップを行ってほしいので，皆さん，Simple-CADをどんどん買ってアンケート葉

表１　部品表

| | | | |
|---|---|---|---|
| センサ | HBCS-1100 | | ¥6,200 |
| OPアンプ | LM311 | | ¥ 100 |
| | LM358 | | ¥ 80 |
| ダイオード | 1SS97 ※2 | @ | ¥ 50 |
| コンデンサ | 0.01μF | | ¥ 15 |
| | 0.47μF ※3 | @ | ¥ 15 |
| 電解コンデンサ | | | |
| | 0.47μF | | ¥ 20 |
| 抵抗 | 100Ω | | ¥ 5 |
| | 1KΩ | | ¥ 5 |
| | 100KΩ | | ¥ 5 |
| | 4.7MΩ | | ¥ 5 |
| | 10MΩ ※2 | @ | ¥ 5 |
| 蛇の目基板 | | | ¥ 100 |
| 9ピンD-SUBコネクタ | | | |
| 3芯ケーブル（50cmもあれば十分） | | | |

図2　バーコードリーダ全回路図



書に「ウィンドウまわりが古くさ〜い」と書いて送り返しましょう。

電気回路以外にも工作の面で気をつけなければならない点があります。センサから4.27±0.25mmの位置にバーコード面が当たるようにしなければなりません。また，バーコード面に対し，センサは垂直でなければなりません。この部分で，バーコードリーダーの精度が決まってしまいますので，工夫が必要です。

私はこのバーコードリーダーを実用的に使う予定はないので，ボールペンのキャップを適当に切り，センサにハメて焦点距離を固定するようにしました。寸法精度はかなりいいかげんです。もし，このシステムでPOSや塗装ロボットのコンピュータ制御などを行う方は，ちゃんとペン型にして垂直がとれるようにするなどの工夫が必要となってくると思います。もしくは，YHPでは，このHBCS-1100を組み込んだペン型センサをコンポーネントとしても発売しているようですので，こちらを使ってもよいでしょう。

## チェックプログラム

リスト1が今回製作したバーコードリーダー用の読み取りプログラムです。

ジョイスティック端子1からは，生の白/黒データが入ってきます。センサからの信号をジョイスティック端子の1番ピンに接続しているので，ジョイスティックでいうところの上方向のボタンが押されたか否かというのがその情報です。上方向に入っていたら(関数stick()の戻り値が8なら)，バーコードリーダーはバーコード面の白いところを指していますし，入っていなければ（同8でないなら）黒いところを指しています。

このデータをバッファリングするのが，関数barCodeGet()です。ここで得られたデータは整数型配列dataに格納されます。白線/黒線，それぞれが何カウント続いたか(つまりは，何モジュール続いたか)が代入され，白黒情報はその要素の添字が奇数だったら黒，偶数だったら白となります。

その次に，その白黒情報は，何モジュールに相当するのかを調べ，正規化するのが関数scale()です。ここで，初めてジョイスティック端子からの入力信号はバーコードの信号と1対1で対応するようになります。

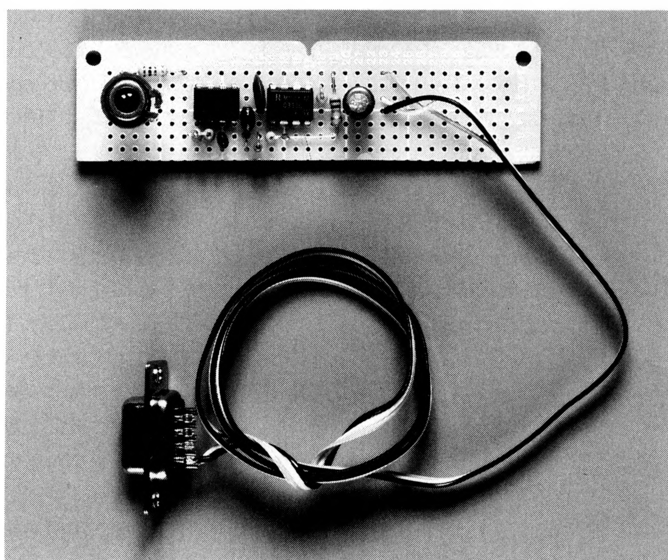
この信号を数値データに解釈するのが関数decode()です。この関数は，引数として16個の要素を持つ整数型配列のポインタを取り，戻り値として，この配列にそれぞれの対応する数値を代入します。この値が−1のときはその値をうまく読めなかったことを表しています。

また，標準コードの場合，0番目，8番目，15番目の要素は数値ではなく，それぞれレフトガードバー，センターバー，ライトガードバーの読み込みに成功したときに1，失敗したときに−1を返します。短縮コードの場合には，6番目，11番目にそれらの情報が入ります。短縮コードの場合にかぎり，例外として0番目の要素には短縮コードであることを示す−2が入ります。

バーコードリーダーは，左から右方向だけでなく，その逆方向からもバーコードデータが読み込まれる可能性があります。また，今回のプログラムでは短縮コードにも対応しましたので，それらの場合分けも行いました。これらの場合分けを言葉で説明してもややこしくなるだけですので，プログラムの構造はプログラム中に埋め込まれた注釈を参照してください。

本来なら，この関数decode()（実際にはこの下請け関数pickUpNum()）は，単純なパターンマッチングを行えばよいハズです。ハズですが，ペン式バーコードの場合，信号の読み込み速度が，手の移動速度に依存するので，一定ではありません。そこで，4本の線を調べてモジュールの和が7にならなかった場合には，そのときの信号は，好意的に解釈してなにを表しているだろうかというのが，配列Under1Num，Under2Num，Under3Num，Over1Numで，それぞれモジュールの和が6, 5, 4, 8だったときのパターンマッチング用の標本です。

で，このように好意的に読み取ったデー



今回作成したバーコードリーダー

図4　LM358とLM311のピンレイアウト

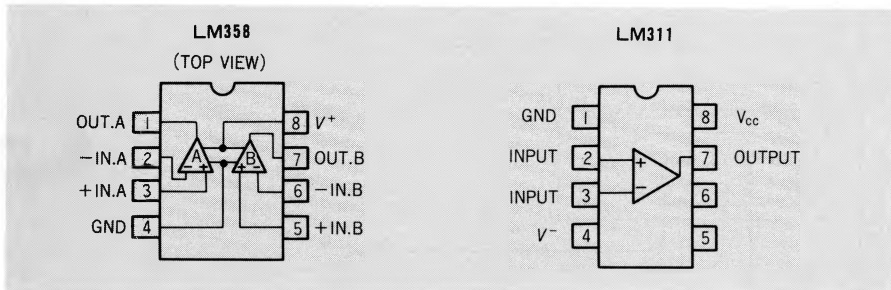



図3-1　基盤のパターン

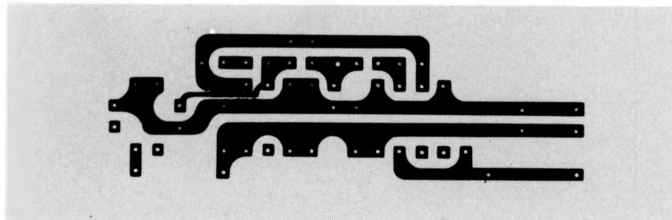
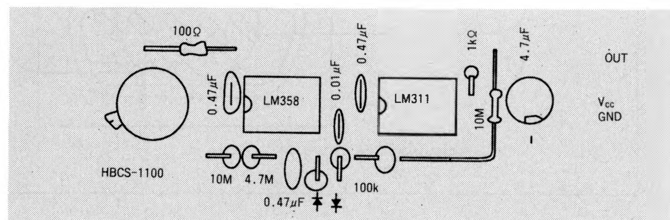


図3-2　実体配線図



タが常に正しいとは限りません。このデータが正しいかを検証するのが関数checksum( )です。JIS X-0501で定められたバーコードは，最左端の数値がモジュロ10というアルゴリズムを用いたチェックサムになっています。各桁に1と3を交互に掛け合わせ，それらの和の一の位が0になればOKです。この検証を行い，正しいコードであるときにchecksum()は1を，そうでないときには−1を返します。

## アプリケーションの作成について

リスト1を改造して，独自のアプリケーションプログラムを作る場合の方法について少し述べましょう。

リスト1中から，まず関数main()を消し去ります。また，読み込んだバーコードデータをグラフィックで表示する必要がない場合には，関数putBarCode()，putBarCode2()もいりません。数値データも表示する必要がない場合には，関数putoutStandard()，putoutShort()もいりません。残りが，バーコードリーダーを使うために最低限必要な関数およびデータです。

データを読み込むには関数barCodeGet()，scale()，decode(BUFF)の順にコールしてやれば，17個の要素を持つ整数型配列BUFFにバーコードのデータが代入されます。詳細は先ほど説明したとおりです。

チェックプログラムを少し動かしてみればわかると思いますが，バーコードリーダーを紙に近づけたり，遠ざけたり（バーコードリーダーにスイッチが付いている場合には）スイッチを入れたり切ったりするたびに，X68000は，そのノイズをバーコードのキャラクタだと誤認してしまいます。これはハードウェアの解説のところで述べたとおり，バーコードの白黒を絶対的な反射率ではなく，相対的な反射率で判定するためです。エラーなくバーコードデータが読み込めるまでループさせておけば，まず問題ないでしょうが，プログラミングの際には頭の片隅に入れておいてください。

## 使い方

では，実際にバーコードリーダーを使ってみます。X68000の電源が入っていない状態で，バーコードリーダーを接続します。接続が完了してから，X68000の電源を入れます。センサ部で用いる光は波長700nmの赤色可視光線ですので，ちゃんと赤色に光っているかを目で確認してください。赤色光が確認されなかったり，X68000が立ち上がらなかったり，変な発振音がしたり，OPアンプが異様に熱かったりした場合は，どこかに配線ミスがありますので，ただちにX68000の電源を切り，回路のチェックを行ってください。

さて，無事X68000が立ち上がれば，次はソフトを実行させます。実行させると，画面左上に“Input Barcode”と出てきますので適当にバーコードにセンサ部分を当て，さっと左から右へ流します。少しして，画面上段に青色で，いま読み込んだ生のデータが，その下に正規化されたデータが表示されます。あなたにバーコード読み取り師としての才能があったり，運がよければ，その次にビープ音が鳴ってバーコードの数値が表示されます。不幸にして，その数字中に「*」や「！」マークが表示された場

### 市販コンポーネントの利用

精度が要求される場合のことも考えて，今回はコンポーネントを使用して，バーコードリーダーをX68000に接続することも行ってみました。写真は沖電気のOPM700BUという製品ですが，プレートが違うだけで中身はYHPのHEDS-3000と同じもののようです。

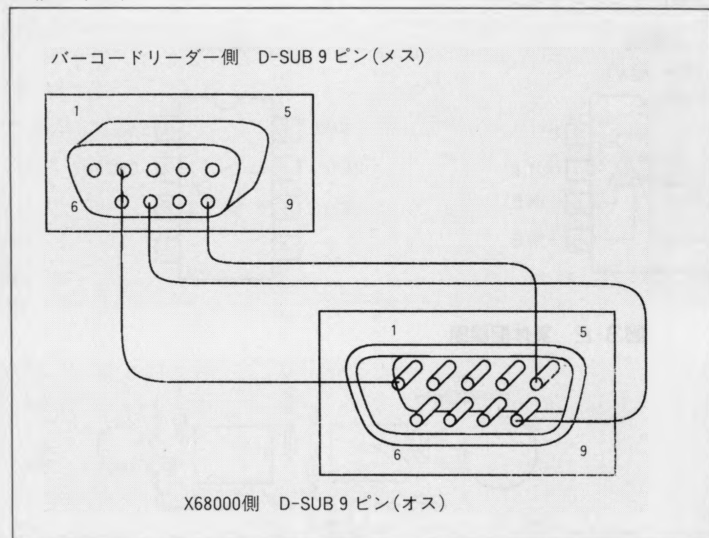
まず，このバーコードリーダーをX68000に接続するには，コネクタを変えてやらなくてはいけません。9ピンD-SUBコネクタは，X68000のジョイスティック端子に差し込むことだけならできるのですが，電源ピンの位置が違うので，ピン配置を変えてやらなければなりません。

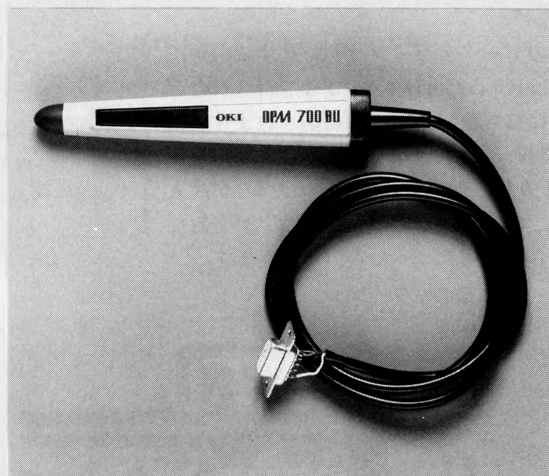
今回は，このコネクタ部分をマイナスドライバで，エイヤッとバラして，9ピンD-SUBコネクタにハンダ付けし直しました。もちろん，そんなことをやればメーカー保証がきかなくなってしまいます。それが嫌な場合には，下図6のような変換コネクタを製作してやる必要があります。

使ってみると，スイッチを押しながらというのがやや不自然な感があるものの，操作感はいたって良好です。洗顔フォームやマヨネーズの包装などに印刷されたバーコードの読み取りには，ちょっと辛いところがあるのですが，コーティングされていない白い紙の上に書かれたバーコードならたいていは一発で読み込めます。さすがに，自作品とは違いメーカーの作ったものだけあるな，と思ったしだいです。ただ，いくらメーカー品といっても人手によってセンサを移動する構造は変わりなく，助走区間がないバーコード（消しゴムやキーホルダーに付いているものなど）を読み取るには，いきなりペンを一定速度で動かし始められるように訓練する必要がありそうです。これは，鍛錬です。

ちなみに，価格のほうは代理店を通して購入すると1万円前後だそうです。

変換コネクタ




OPM700BUによる製作例

合は，それぞれ数値データ，ガイドバーの読み込みミスですので，どこが悪かったのかを画面を見ながら点検し，バーコードが読めるように練習してください。「！」や「＊」が表示されているのにビープ音が鳴らない場合には，どこかの数値が間違っています。もう一度，トライしてください。

ひと通りの練習が終わって，プログラムを終了させたいときには，BREAKキーを押すかCTRLキーとCを共に押してから，適当なバーコードを読み込ませます（つまりは，専用の処理を行っていないので，標準出力に書き込みにいくときのBREAKキーチェックを利用している）。




## これから

本当なら，ここから「バーコードバトラー」を作ると面白いのですが，今回はそこまで手がまわりませんでした。もっとも，今回のリーダーの読み取り率では，そんなことを行う前に手が疲れてしまいそうです。ここから先の読み取り率の向上，「バーコードバトラー」やスキャナ型センサは読者の方々の課題としましょう。

**参考文献**


1)　流通システム開発センター編，バーコードのおはなし，日本規格協会
2)　池田　央，バーコードリーディングシステムの設計と製作，トランジスタ技術1990年8月号，ＣＱ出版
3)　HBCS-1100データシート，横河ヒューレット・パッカード
4)　中野修一，バーコードバトラーの解析，Oh!X1992年4月号
5)　JISX-0501共通商品コード用バーコードシンボル，JISハンドブック


**リスト1：GET.C**

```
================== get.c ==================
/*
**          バーコード読み取りプログラム
**                  JIS-X-0501
**
**                  Programed by 石上 達也
**                          07/20/92
**
*/

#include  <stdio.h>
#include  <stick.h>

#define   BUF_SIZE        60
#define   WHITE           8
#define   BLACK           0
#define   SHORT           -2

/*
**        データキャラクタのパターン
*/
int       Over1Num[] = {/*バーを1モジュール太くする*/
  0,      /* 00 */
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  4112,   /* 90 */

  2114,   /* 0E */
  1223,
  0,
  0,
  0,
  1421,
  2141,
  0,
  0,
  2114    /* 9E */
};

int JustNum[] = {              /*標準モジュールデータ*/
  3211,             /* 00 */
  2221,
  2122,
  1411,
  1132,
  1231,
  1114,
  1312,
  1213,
  3112,             /* 90 */

  1123,             /* 0E */
  1222,
  2212,
  1141,
  2311,
  1321,
  4111,
  2131,
  3121,
  2113              /* 9E */
};

int Under1Num[] = { /* 最太バーを1モジュール細くする */
  2211,    /* 00 */
  0,
  0,
  1311,
  0,
  0,
  1113,
  0,
  0,
  2112,    /* 90 */

  0,       /* 0E */
  0,
  0,
  1131,
  0,
  0,
  3111,
  0,
  2121,
  0        /* 9E */
};

int Under2Num[] = { /* 最太バーを2モジュール細くする */
  0,       /* 00 */
  1211,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,       /* 90 */

  1112,    /* 0E */
  0,
  0,
  0,
```

```
  0,
  0,
  0,
  0,
  0,
  0        /* 9E */
};

int Under3Num[] = { /* 最太バーを3モジュール細くする */
  0,       /* 00 */
  1111,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,       /* 90 */

  0,       /* 0E */
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0,
  0        /* 9E */
};

/*
**プリフィックスのデータ
*/
char       *masterPrefix[] = {
  "OOOOOO",
  "OOEOEE",
  "OOEEOE",
  "OOEEEO",
  "OEOOEE",
  "OEEOOE",
  "OEEEOO",
  "OEOEOE",
  "OEOEEO",
  "OEOEEO",
  "OEEOEO"
};

int        data[BUF_SIZE];

main() {
int        i;
int        num[17];

  screen(2,0,1,1);

  while(1) {
          puts("Input Bar-Code Data!");
          barCodeGet();
          putBarCode();
          scale();
          putBarCode2();
          decode(num);
          if(checksum(num) == 1)/*読み取りに成功したらベルを鳴らす*/

                  putchar(7);
          if(num[0] != SHORT) {
                  putoutStandard(num);
          } else {
                  putoutShort(num);
          }
  }
}

/*
  標準コードと見なして画面に表示
*/
putoutStandard(int num[])
{
  int     i;

  for(i=0; i<=15 ; i++) {
          if(i == 1 || i == 8 || i == 15) {
                  if(num[i] == 1) putchar('|');
                  else            putchar('!');
          } else {
                  if(num[i] >= 0) printf("%d",num[i]);
                  else            putchar('*');
          }
  }
  putchar('¥n');
}

/*
  短縮コードと見なして画面に表示
*/
putoutShort(int num[])
```

```
{
  int     i;

  for(i=1; i<=11 ; i++) {
          if(i == 1 || i == 6 || i == 11) {
                  if(num[i] == 1) putchar('|');
                  else            putchar('!');
          } else {
                  if(num[i] >= 0) printf("%d",num[i]);
                  else            putchar('*');
          }
      }
  /*画面から標準コードの文字を消す*/
  printf("     ¥n");
}

/*
  生のデータを画面に表示する
*/
putBarCode()
{
  int     i;
  int     newx, oldx, color;

  wipe();
  newx = oldx = 0;
  color = 3;
  for(i = 0; i <= BUF_SIZE; i++) {
          newx += data[i] / 20;
          fill(oldx,0,newx,100,color);
          oldx = newx;
          color = color ^ 3;
  }
}

/*
  正規化されたデータを画面に表示する
*/
putBarCode2()
{
  int     i;
  int     newx, oldx, color;

  newx = oldx = 0;
  color = 5;
  for(i = 0; i <= BUF_SIZE; i++) {
          newx += data[i] * 3;
          fill(oldx,100,newx,200,color);
          oldx = newx;
          color = color ^ 5;
  }
}

/*
  バーコードリーダからデータを読み取る
*/
barCodeGet()
{
  int     i, cnt, last;
  int     trap = BLACK;

  for(cnt = 0; cnt < BUF_SIZE; cnt++ ){
          data[cnt] = 0;
  }

  while(stick(1) != WHITE);       /*センサーを紙につけた*/

  while(stick(1) != BLACK);       /*Start Bar */

  last = i = 0;
  for(cnt = 0; cnt < 0x8ffffff ; cnt++) {
          cnt++;
          if(stick(1) == trap) {
                  data[i++] = cnt - last;
                  trap = trap ^ 8;
                  last = cnt;
                  if(i >= BUF_SIZE) break;
          }
          if(cnt - last > 10000) break;   /* time Up*/
  }
  return(cnt);
}

/*
  データを正規化する
*/
int scale()
{
  int     i;
  int     vel;              /*1ビッチの長さ(可変)*/

  if((vel = data[1]) == 0) {
          return(-1);
  }
  for(i = 1; data[i] && i <= BUF_SIZE; i++) {
          if(data[i] < vel*1+vel/2)  {
                  vel = data[i];
                  data[i] = 1;
```



▶「オーバーテイク」のデモ&ゲームはすごい。実写アニメーションはすごく滑らかでたまらなくいい。ああ、よかった。
藤巻 康昌(19)栃木県

```
		} else if(data[i] < vel*2+vel/2) {
			vel = data[i]/2;
			data[i] = 2;
		} else if(data[i] < vel*3+vel/2) {
			vel = data[i]/3;
			data[i] = 3;
		} else {
			vel = data[i]/4;
			data[i] = 4;
		}
	}
	return(1);
}

/*
	正規化されたデータを解読する
*/
decode(int ret[])
{
	int	num;
	char	prefix[6];

	ret[1] = checkGuardbar(1); /*ガードバーのチェック*/
	if(getLeftNum(4) >= 10) {  /*いきなり偶数パリティということは*/
					/*逆向き*/
		ret[15] = ret[1];   /*さっきのガードバーは最左端バー*/
		decodeBackward(ret, prefix);
	} else {
		decodeForward(ret, prefix);
	}
}

/*
	順方向に解読
*/
decodeForward(int ret[], char prefix[])
{
	decodeFirst4(ret, prefix); /*とりあえず初めの4文字を読み込む*/

	if(checkCenterbar(20) == 1) { /*5文字目がセンターバーなら*/
		decodeShort(ret, prefix);		/*短縮コード*/
		ret[0] = SHORT;
	} else {				/*そうでないなら*/
		decodeStandard(ret, prefix);	/*標準コード*/
	}
}

/*
	とりあえず初めの4文字を読み込む
*/
decodeFirst4(int ret[], char prefix[])
{
	int	ptr = 4;
	int	cnt;
	int	num;

	for(cnt = 0; cnt <= 3; cnt++) {
		num = getLeftNum(ptr);
		ptr += 4;
		if(num >= 10) {
			num = num - 10;
			prefix[cnt] = 'E';
		} else {
			prefix[cnt] = 'O';
		}
		ret[cnt + 2] = num;
	}
}

/*
	標準コードの残りを読み取る
*/
decodeStandard(int ret[], char prefix[])
{
	int	num, cnt;
	int	ptr = 20;

	for(cnt = 4; cnt <= 5; cnt++) {
		num = getLeftNum(ptr);
		ptr += 4;
		if(num >= 10) {
			num = num - 10;
			prefix[cnt] = 'E';
		} else {
			prefix[cnt] = 'O';
		}
		ret[cnt + 2] = num;
	}
	ret[8] = checkCenterbar(ptr);
	ptr += 5;
	for(cnt = 6; cnt <= 11; cnt++) {
		ret[cnt + 3] = getRightNum(ptr);
		ptr += 4;
	}

	ret[15] = checkGuardbar(ptr);

	ret[0] = getFirstNum(prefix);
}

/*
	短縮コードの残りを読み取る
*/
decodeShort(int ret[], char prefix[])
{
	int	num, cnt;
	int	ptr = 25;

	if(strncmp(prefix, "OOOO", 4)) {
		/*右側に偶数パリティコードが混ざっている時*/
		/*短縮コードではない*/
		ret[6] = -1;
	} else {
		ret[6] = checkCenterbar(ptr);
		for(cnt = 4; cnt <= 7; cnt++) {
			ret[cnt + 3] = getRightNum(ptr);
			ptr += 4;
		}
		ret[11] = checkGuardbar(ptr);
	}
}

/*
	逆方向に解読
*/
decodeBackward(int ret[], char prefix[])
{
	decodeFirst4B(ret);	/*とりあえず初めの4文字を読み込む*/

	if(checkCenterbar(20) == 1) { /*5文字目がセンターバーなら*/
		ret[11] = ret[15];		/*左に詰めて*/
		ret[10] = ret[14];
		ret[ 9] = ret[13];
		ret[ 8] = ret[12];
		decodeShortB(ret);		/*短縮コード*/
		ret[0] = SHORT;
	} else {				/*そうでないなら*/
		decodeStandardB(ret, prefix);	/*標準コード*/
	}
}

/*
	とりあえず初めの4文字を読み込む(逆方向)
*/
decodeFirst4B(int ret[])
{
	int	ptr = 4;
	int	cnt;

	for(cnt = 14; cnt >= 11; cnt--) {
		ret[cnt] = getRightNumB(ptr);
		ptr += 4;
	}
}

/*
	標準コードの残りを読み取る(逆方向)
*/
decodeStandardB(int ret[], char prefix[])
{
	int	num, cnt;
	int	ptr;

	ret[10] = getRightNumB(20);
	ret[9] = getRightNumB(24);
	ret[8] = checkCenterbar(28);

	ptr = 33;
	for(cnt = 7; cnt >= 2; cnt--) {
		num = getLeftNumB(ptr);
		if(num >= 10) {
			num = num - 10;
			prefix[cnt-2] = 'E';
		} else {
			prefix[cnt-2] = 'O';
		}
		ptr += 4;
		ret[cnt] = num;
	}

	ret[1] = checkGuardbar(ptr);

	ret[0] = getFirstNum(prefix);
}

/*
	短縮コードの残りを読み取る(逆方向)
*/
decodeShortB(int ret[])
{
	int	num, cnt;
	int	ptr = 25;

	ret[6] = checkCenterbar(ptr);
	for(cnt = 5; cnt <= 2; cnt--) {
		ret[cnt] = getLeftNumB(ptr);
```

▶「ムーンクレスタ」。私が小学生のときかなりやり込んだ覚えがあります(音楽まではっきり記憶にあります)。ところが、友人のほとんどが「ムーンクレスタ」を知らないのです。きっと「スクランブル」も知らないんだろうな。う～ん。　陣山　達夫(22)大阪府

```
                ptr += 4;
    }
    ret[1] = checkGuardbar(ptr);
}

/*
    左側のキャラクタデータを順方向に読み取る
*/
getLeftNum(int ptr)
{
    int     d1,d2,d3,d4,sum;
    int     i;

    d1 = data[ptr];
    d2 = data[ptr+1];
    d3 = data[ptr+2];
    d4 = data[ptr+3];

    sum = d1 + d2 + d3 + d4;
    i = d1*1000 + d2*100 + d3*10 + d4;
    return(pickUpNum(sum, i));
}

/*
    左側のキャラクタデータを逆方向に読み取る
*/
getLeftNumB(int ptr)
{
    int     d1,d2,d3,d4,sum;
    int     i;

    d1 = data[ptr];
    d2 = data[ptr+1];
    d3 = data[ptr+2];
    d4 = data[ptr+3];

    sum = d1 + d2 + d3 + d4;
    i = d1 + d2*10 + d3*100 + d4*1000;
    return(pickUpNum(sum, i));
}

/*
    右側のキャラクタデータを順方向に読み取る
    (ということは、裏返された偶数パリティの
    キャラクタを順方向に読み取ることに他ならない)
*/
getRightNum(int ptr)
{
    int     d1,d2,d3,d4,sum;
    int     i;

    d1 = data[ptr];
    d2 = data[ptr+1];
    d3 = data[ptr+2];
    d4 = data[ptr+3];

    sum = d1 + d2 + d3 + d4;
    i = d1 + d2*10 + d3*100 + d4*1000;          /*裏返し*/

    i = pickUpNum(sum , i);
    if(i < 10) {
    /*奇数パリティのデータが読み込まれてはいけない*/
            return(-1);
    } else {
    /*奇数/偶数パリティの判定はもういらないので*/
    /*10を引いてやる                          */
            return(i-10);
    }
}

/*
    右側のキャラクタデータを逆方向に読み取る
    (ということは、順方向に偶数パリティの
    キャラクタデータだけを読み取ればよい)
*/
getRightNumB(int ptr)
{
    int     d1,d2,d3,d4,sum;
    int     i;

    d1 = data[ptr];
    d2 = data[ptr+1];
    d3 = data[ptr+2];
    d4 = data[ptr+3];

    sum = d1 + d2 + d3 + d4;
    i = d1*1000 + d2*100 + d3*10 + d4;          /*順方向*/

    i = pickUpNum(sum, i);

    if(i < 10) {
    /*奇数パリティのデータが読み込まれてはいけない*/
            return(-1);
    } else {
    /*奇数/偶数パリティの判定はもういらないので*/
    /*10を引いてやる                          */
            return(i-10);
```

```
    }
}

/*
    下請け関数
*/
pickUpNum(int sum, int i)
{
int     j;
    if(sum ==  4) {
            for(j=0; j <= 19; j++) {
                    if(i == Under3Num[j]) return(j);
            }
    } else if(sum ==  5) {
            for(j=0; j <= 19; j++) {
                    if(i == Under2Num[j]) return(j);
            }
    } else if(sum ==  6) {
            for(j=0; j <= 19; j++) {
                    if(i == Under1Num[j]) return(j);
            }
    } else if(sum == 7) {
                    for(j=0; j <= 19; j++) {
                    if(i == JustNum[j]) return(j);
            }
    }  else if(sum == 8) {
            for(j=0; j <= 19; j++) {
                    if(i == Over1Num[j]) return(j);
            }
    }
    return(-1);
}

/*
    センターバーの確認
*/
int checkCenterbar(int ptr)
{
    if((data[ptr++] == 1)
    && (data[ptr++] == 1)
    && (data[ptr++] == 1)
    && (data[ptr++] == 1)
    && (data[ptr]   == 1))
            return(1);
    else
            return(-1);
}

/*
    ガードバーの確認
*/
checkGuardbar(int ptr)
{
    if((data[ptr++] == 1)
    && (data[ptr++] == 1)
    && (data[ptr]   == 1))
            return(1);
    else
            return(-1);
}

/*
    プリフィックスデータを読み取る
*/
int getFirstNum(char prefix[])
{
    int     i;

    for(i = 0; i < 10; i++) {
            if(strncmp(prefix, masterPrefix[i], 6) == 0)
                    return(i);
    }
    return(-1);
}

/*
    読み込まれたバーコードをチェックする
    (OK = 1, NG = -1)
*/
checksum(int num[])
{
    int     i;

    if(num[0] != SHORT) {    /*標準コード*/
            i =   num[0]+num[3]+num[5]+num[7]+num[10]+num[12]+num[14];
            i += (num[2]+num[4]+num[6]+num[9]+num[11]+num[13])*3;
            i = i % 10;
    } else      {               /*短縮コード*/
            i =  num[2]+num[4]+num[7]+num[9];
            i += (num[3]+num[5]+num[8]+num[10])*3;
            i = i % 10;
    }
    if(i == 0) return(1);
    else return(-1);
}
```



(で)のショートプロぱーてぃ――その40

# きっちり揃わぬ隙間風

Komura Satoshi 古村 聡

木枯らしピューピュー隙間風，こんな日はコタツでぬくぬくプログラミングかな。今月は，ちょっと強引だけどIOCSのスプライト関連命令を高速化する「SP.S」，楽々字詰めの「ADJUST.C」をお届けしましょう。



illustration : T. Takahashi

皆さん，あけましておめでとう。お元気ですか，(で)でございます。私めはまたしても風邪をひいてしまいました。なんだか，冬になると毎年こんなことを書いている自分が情けないです，はい。

これは，現在私の住んでいるバストイレ給湯設備付き，冷暖房完備，床は総フローリングと結構「ごうせい」な部屋がいけないんです。この部屋に引っ越してきたときに，気がついたのですが，この部屋，扉がすごく重くて，私でも両手で思いっきり押さないと扉が開かなかったんですよ。このときは管理人さんにいったら，すぐに，ドアの枠を削って対処してくれた(おいおいおい)ので扉は簡単に開くようになったのです。ところが，たてつけのおかしかったのは扉だけではなかったのですね。

冬になって少し寒くなってから気がついたのですが，サッシ窓に隙間があって完全に閉まらないのですよ。もう，寒くって，寒くって。今度は管理人さんも「サッシ窓はどうしようもないので，新聞紙かガムテープで隙間を埋めてくれ」だって……。

いまではX68000と一緒に毛布にくるまって原稿を書いているのです(X68000って結構熱持つし，ファンが風出すしね)。ううっ，年の初めからひとりさびしく，X68000抱えているなんて，そんなみじめな青春，絶対にいやじゃぁ。ぐっすし。

##  ゴリゴリ削れIOCS

では，今月の1本目にいきましょう。まず，1本目のプログラムは愛知県の神山さんによる作品で，どうしても作ったゲームが速くならないときの切り札ツール，SP.Sです。

**SP.S for X68000**

**(要AS.X, LK.X, DOSCALL.MAC)**

**愛知県 神山満**

これは，X-BASICで書いてコンパイルしたゲームなどのプログラムにある，スプライト関係の命令の実行を高速化してしまうプログラムSP.Sです。

このプログラムのリストはアセンブラのソースリストの形で書かれていますので，AS.X，LK.X，それに定義ファイルDOSCALL.MACが必要です。また，アセンブラはVer.2.0以上のものを使ってください(フリーウェアのHAS.Xなどでもかまいません)。アセンブル方法は，

```
A>AS SP.S
A>LK SP.O
```

で実行ファイルを作ります。

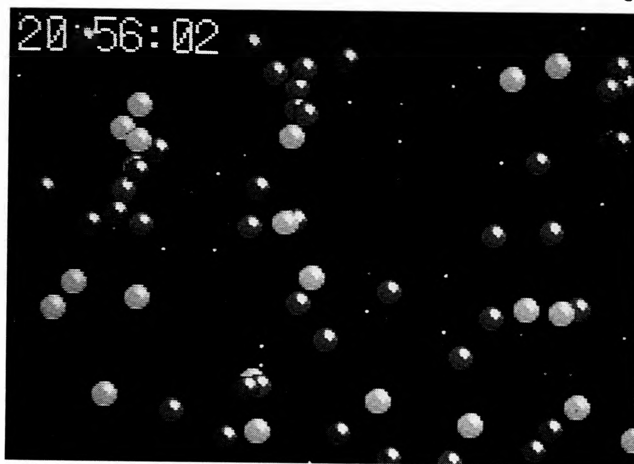
さて，実際にスプライトが速くなっているか試してみましょう。リスト2のSPTEST.BASをコンパイルしてみてください(SP.Xを常駐させたまま，インタプリタ上で実行しないようにね)。それから，sptest.xを実行してみて，次にsp.xを組み込んでからsptest.xを実行しましょう。

テストプログラムを，SP.Xが常駐しているときとしていないときで比べると，10%ぐらい高速化されているのがわかると思います。また，純粋にスプライト関係の命令だけに限ると15%ほど速くなっているはずです。どうです，違いがわかりますか(なんだか，洗剤のコマーシャルみたいだな)。

実はこのプログラム，「BASICコンパイラでコンパイルした実行ファイルだけを速くする」ので，ほかのプログラム，たとえば，ほとんどのCやアセンブラで作ったプログラムや市販のゲームなどで，スプライトレジスタを直接自分でアクセスしているプログラムには，まったく対応していません。

というのもこのプログラムは，元のIOCSルーチンからレジスタの保存やエラーチェック，必要のない戻り値の設定，垂直同期のフラグのチェックなどを取り払うことで高速化を行っているからなのです。ゲーム中リアルタイムで使用される関数は，sp_move，sp_set，sp_color，bg_put，bg_scrollあたりですから，それに対応したIOCSコールを書き換えることでスプライトの高速化を達成しているのです(X-BASICの関数も結局はIOCSコールを呼び出してますからね)。

というわけですから，絶対にX-BASICインタプリタでは使わないでください。そして，二重常駐はしないようにチェックを



SP.S

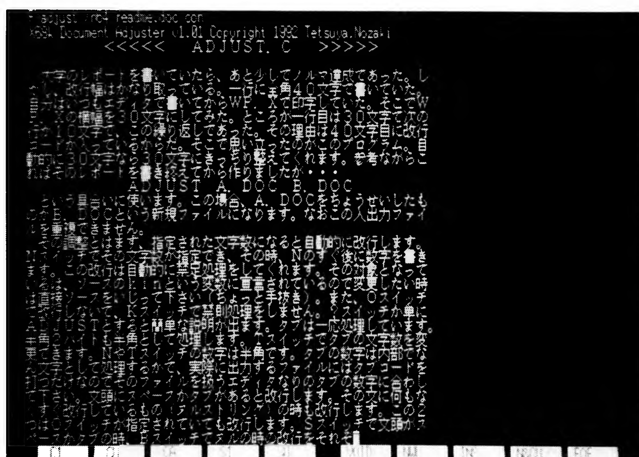
していますが，常駐解除はできません。ここにも注意してください。

このプログラム，方法としてはとても強引ですから，使うときには十分注意して使わなくてはいけないんですよね。うーん，それを考えると，やはり常駐解除はつけてほしかったなぁ。次はお願いしますね，神山さん。よろしく。



## 38列縦隊ADJUST.C

では，今月の２本目いきます。大阪府は野崎さんのプログラムでADJUST.Cです。どーぞ。



ADJUST.C

### ADJUST.C for X68000

**(要Cコンパイラ)**

**大阪府　野崎哲也**

「大学のレポートを書いていたら，あと少しでノルマ達成であった。しかし，改行幅はかなり取っている。１行に全角40文字で書いていた。自分はいつもエディタで書いてからWP.Xで印字していた。そこでWP.Xの横幅を30文字にしてみた。ところが一行目は30文字で次の行が10文字で，この繰り返しであった。その理由は40文字目に改行コードが入っているからだ」

というような野崎さんの経験から作成されたのが，改行コードを一定の間隔でつけ直してくれる字詰めプログラム，ADJUST.Cです。これがあれば，どんな改行をしていても自動的に30文字なら30文字に，きっちり整えてくれます。

このプログラムはCで書かれていますので，

A>CC ADJUST.C

でコンパイルしてください。標準ライブラリのみを使っているのでオプションスイッチはいりません(お好みで最適化スイッチ/Oなどをつける分にはいっこうにかまいませんけど)。

そして，実行ファイルADJUST.Xができましたら，

A>ADJUST A.DOC B.DOC

という具合に使います。この場合，A.DOCというドキュメントファイルの字詰めを調整したものがB.DOCという名前で出力されます。この入出力ファイル名は同じにすることはできません。

ところでその字詰めの調整ですが，/Nスイッチでその文字数が指定できます。Nのすぐあとに半角で数字を書いてください。その数字は半角文字数ですので注意が必要です。具体例として，

んー。いい天
気だねぇ。さぼ
っちゃいたいなー。

という文章を/N28で字詰めを行うと，

んー。いい天気だねぇ。さぼっ
ちゃいたいなー。

としてくれるわけです。

また，この改行プログラムは自動的に禁則処理をしてくれます。その対象となっている文字は，ソースのkinという変数に宣言されているので変更したいときは直接ソースをいじってください。

また，/Oスイッチで改行なし，/Kスイッチで禁則処理の禁止です。また，/?スイッチか単に，

A>ADJUST

で簡単な使い方の説明が表示されます。

タブコードも一応処理しています。/Tスイッチでタブの文字数を変更できますが，この数字はタブコードをスペース何文字分とみなすかだけで，実際に出力するファイルにはタブコードを打つだけです。

あと，/Sスイッチで文頭にスペースかタブがあると改行します。これは/Oスイッチが指定されていても改行します。そして，/Bスイッチでその文のなにもないヌル行の改行を禁止します。……ひあー，もはや常連と化した感のある野崎さんだけあって，作るプログラムもすごいすごい。この短さでこのオプションスイッチの山。

実をいうと，今月の原稿を書くときにもたっぷり活用させていただきました。この原稿を書くとき，改行のたくさん入った投

## 動かないよ，と思う前に(3)

今月のリストはCとアセンブラですけど，まぁ，いままでの２回の解説で，Cもアセンブラも電話がかかってきそうなものは，だいたい解説してしまったのでもうあまり迷うところもないでしょうね。

今月は間違えないリストの打ち込み方，なんてのを伝授してしまいましょう。

**●１行飛ばしてしまった**

素直に打ち込みやっているとよくやるミスがこれ。特にCのリストのときに多いのですが，本を見ながら打ち込みをやっているとついつい１行飛ばして入力してしまうことが多いようです。せっかくOh!XではCのリストにも１行１行行番号をふっているのですからぜひとも活用してください。また１行１行番号をふっていても注釈行を入力するときに飛ばしてしまえば，なんにもならないですね。結構，ありがちなパターンなのです。気をつけましょう。

**●Warningに気をつけて**

で，アセンブラやCにありがちな話なのですが，打ち込みミスの場合にはエラーメッセージではなくWarning（警告）メッセージが出ることもあります。プログラマがわかっている分には，基本的に無視していいエラーなのですが，打ち込みなどでWarningメッセージが出て，そのプログラムが正常に動作しない場合は注意が必要です。たとえば，

```
move.w #$3000,a0
```

と書くべきところを，

```
move.w $3000,a0
```

と書いてしまうとエラーではなくWarningでアセンブラはリストを通してしまうのですが，やっぱりこの場合はおかしな動きをしてしまうのです。おかしいな，と思ったら，Warningの内容も疑ってみてくださいね。

**●で，やっぱり見直し！**

で，やっぱり最後にいちばん頼りになるのは見直しです。何度もいうようですが，エラーメッセージやWarningメッセージが出たり，なにかおかしな動作をしたとき，メッセージに出ている，その行が違っているとは限らないのです。その前後する行が違っていたり，あるいはまったく離れた行がおかしいということは応々にしてあるのです。なにかおかしいと思ったらまずリストを見直してみましょう。その行の前後から全体を見るようにしてみてください。

でもって，ほんとになんか変だな，と思ったら編集室に電話してみてくださいね。

稿原稿をワープロで見るのにとても便利でしたよん。いつもは改行コードのあるまま見てたんですよね。野崎さん，えらいっ！

さて，この調子で誤字脱字のチェックをしてくれるプログラムとか，文章を推敲してくれるプログラムとか，ついでに投稿原稿から勝手にショートプロの原稿を書いてくれるプログラムとか，じゃんじゃん送られてくるとうれしいんですけどね。

え，なに，勝手にショートプロができてくるならお前なんかいらないって。締め切り破りも誤字脱字もないし，そんな殺生な，編集さん……。

うー，それでは失業に気をつけつつまた来月。こんな不景気の世の中で路頭に迷うのはヤだぞ，わし。




## リスト1 SP.S

```
*=====================================
*                  SP.S
*
*    COPYRIGHT 1987 SHARP/Hudson
*    COPYLEFT  1992 Mitsuru Kamiyama
*=====================================

          .include        doscall.mac

top:      bra     entry

*SP_REGST ($C6)
c6:       lea.l   $00eb0000,a0
          lsl.w   #3,d1
          adda.w  d1,a0
reg_x:    tst.l   d2
          bmi     reg_y
          move.w  d2,(a0)
reg_y:    addq.w  #2,a0
          tst.l   d3
          bmi     reg_pc
          move.w  d3,(a0)
reg_pc:   addq.w  #2,a0
          tst.l   d4
          bmi     reg_pr
          move.w  d4,(a0)
reg_pr:   addq.w  #2,a0
          tst.l   d5
          bmi     reg_rt
          move.w  d5,(a0)
reg_rt:   rts

*BGSCROLL ($C8)
c8:       btst.l  #0,d1
          bne     bgs_1
          lea.l   $00eb0800,a0
          bra     bgs_x
bgs_1:    lea.l   $00eb0804,a0
bgs_x:    tst.l   d2
          bmi     bgs_y
          move.w  d2,(a0)
bgs_y:    addq.w  #2,a0
          tst.l   d3
          bmi     bgs_rt
          move.w  d3,(a0)
bgs_rt:   rts

*BGTEXTST ($CD)
cd:       btst.l  #0,d1
          bne     bgt_1
          lea.l   $00ebc000,a0
          bra     bgt_wt
bgt_1:    lea.l   $00ebe000,a0
bgt_wt:   add.w   d2,d2
          lsl.w   #7,d3
          add.w   d3,d2
          adda.w  d2,a0
          move.w  d4,(a0)
          rts

*SPALET ($CF)
cf:       lea.l   $00e82200,a0
          add.w   d1,d1
          lsl.w   #5,d2
          add.w   d2,d1
          adda.w  d1,a0
          tst.l   d3
          bpl     spa_wt
          moveq.l #0,d0
          move.w  (a0),d0
          rts
spa_wt:   moveq.l #0,d0
          move.w  (a0),d0
          move.w  d3,(a0)
          rts

*-------------------------------------

entry:    pea     mes(pc)
          DOS     _PRINT
          addq.l  #4,sp

          move.w  #$1c6,-(sp)
          DOS     _INTVCG
          addq.l  #2,sp
          andi.l  #$ff0000,d0
          cmpi.l  #$ff0000,d0
          bne     err

          pea     c6(pc)
          move.w  #$1c6,-(sp)
          DOS     _INTVCS
          pea     c8(pc)
          move.w  #$1c8,-(sp)
          DOS     _INTVCS
          pea     cd(pc)
          move.w  #$1cd,-(sp)
          DOS     _INTVCS
          pea     cf(pc)
          move.w  #$1cf,-(sp)
          DOS     _INTVCS
          lea     24(sp),sp

          clr.w   -(sp)
          move.l  #entry-top,-(sp)
          DOS     _KEEPPR
          addq.l  #6,sp

err:      pea     errmes(pc)
          DOS     _PRINT
          addq.l  #4,sp
          DOS     _EXIT

          .data

mes:      dc.b    'スプライト/BG高速化 for XBAStoX ver1.00',13,10
          dc.b    '(C)1987,1992 SHARP/Hudson/Mitsuru Kamiyama',13,10,0
errmes:   dc.b    13,10
          dc.b    'すでに組み込まれています',13,10,0

          .end
```

## リスト2 SPTEST.BAS

```
 10 /* スプライトテスト
 20   int i,j,wk,wkx,wky,wc,bg0x=0,bg1y=0
 30  dim int ball(5*100)
 40  dim char bchr(63)
 50  dim int pal_dat(16*3)={
 60   +0,21140,64,384,768,1280,1664,1792,1984,8134,
 70   +16334,24534,34784,45034,57334,65534,
 80   +0,20,21140,32,46,10606,12734,19070,21182,25406,
 90   +25406,31742,46526,52862,61310,65534,
100   +0,21140,32,26624,32900,49482,59986,64346,64544,
110   +64610,64808,64940,65138,65270,65402,65534 }
120  dim char c(255)={
130   +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
140   +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
150   +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
160   +0,0,0,0,0,0,6,8,11,9,0,0,0,0,0,0,
170   +0,0,0,0,0,6,8,11,13,13,9,0,0,0,0,0,
180   +0,0,0,0,5,8,8,11,15,15,13,0,0,0,0,0,
190   +0,0,0,0,7,8,8,8,14,15,13,9,0,0,0,0,
200   +0,0,0,0,7,8,8,8,8,14,11,11,0,0,0,0,
```

```
210     +0,0,0,0,7,7,8,8,8,11,10,8,0,0,0,0,
220     +0,0,0,0,6,7,8,8,8,8,8,8,0,0,0,0,
230     +0,0,0,0,5,7,7,7,8,8,8,5,0,0,0,0,
240     +0,0,0,0,0,6,7,7,7,7,7,0,0,0,0,0,
250     +0,0,0,0,0,4,6,7,7,6,4,0,0,0,0,0,
260     +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
270     +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
280     +0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 }
290    screen 0,0,1,0:sp_init()
300    print time$:all_init()
310      for i=0 to 10000
320        ball_move()
330      next:print time$:print"push any key!"
340    sp_disp(0):while inkey$="":endwhile
350    end
360  func ball_move()
370    int i
380     for i=0 to 99
390      wk=i*5
400      wkx=ball(wk)+ball(wk+2):wky=ball(wk+1)+ball(wk+3)
410       if wkx>240 then ball(wk+2)=-1:wkx=wkx-1
420       if wkx<0 then ball(wk+2)=1:wkx=wkx+1
430       if wky>240 then ball(wk+3)=-1:wky=wky-1
440       if wky<0 then ball(wk+3)=1:wky=wky+1
450        ball(i*5)=wkx:ball(wk+1)=wky
460        sp_set(i,ball(wk)+16,ball(wk+1)+16,ball(wk+4),3)
470      next
480       bg0x=bg0x+2:bg1y=bg1y+2
490       if bg0x>511 then bg0x=0
500       if bg1y>511 then bg1y=0
510       bg_scroll(0,bg0x,0):bg_scroll(1,0,bg1y)
520      /* 余計な命令
530    sp_def(0,c)
540    for i=1 to 15:bchr(0)=i:sp_def(5+i,bchr,0):next
550  endfunc
560  func all_init()
570    sp_def(0,c)
580    for j=1 to 3
590     for i=0 to 15
600      sp_color(i,pal_dat((j-1)*16+i),j)
610     next:next
620       for i=0 to 99
630         wk=i*5
640         ball(wk)=int(1000*rnd()) mod 240
650         ball(wk+1)=int(1000*rnd()) mod 240
660         ball(wk+2)=(int(10*rnd()) mod 3)-1
670         ball(wk+3)=(int(10*rnd()) mod 3)-1
680         ball(wk+4)=((int(10*rnd()) mod 3)+1)*256
690         sp_set(i,ball(wk)+16,ball(wk+1)+16,ball(wk+4),3)
700        next
710    bg_set(0,0,1):bg_set(1,1,1)
720    bg_fill(0,5):bg_fill(1,5)
730      for i=0 to 63:bchr(i)=0:next
740      for i=1 to 15:bchr(0)=i:sp_def(5+i,bchr,0):next
750     for j=0 to 1
760      for i=0 to 200
770         wkx=int(100*rnd()) mod 64
780         wky=int(100*rnd()) mod 64
790         wk=(int(100*rnd()) mod 15)+6
800         wc=((int(10*rnd()) mod 3)+1)*256+wk
810        bg_put(j,wkx,wky,wc)
820      next:next
830    sp_disp(1):sp_on(0,127)
840  endfunc
```

## リスト3 ADJUST.C

```c
/*****************************************************************
                文章調整プログラム Ver 1.01
                                平成4年9月26日
                        Programmed By 野崎 哲也
                        Arranged   By (て)
                        (11月20日 /b,禁則,変数 endをvendに)
******************************************************************/
#include <stdlib.h>
#include <stdio.h>

FILE *motoFile;
FILE *sakiFile;
char *motoName = NULL;
char *sakiName = NULL;
int num = 80;
int moji;
int start;
int zenFlag;
int c1,c2;
int vend = 0;
int nullFlag = 0;
int spaceFlag = 0;
int kinsokuFlag = 0;
int tab = 8;
unsigned char kin[]="、。，．？！゛゜'）〕］｝〉》」』】＞～’”";

void manual( void )
{
 puts("使用法:adjust [スイッチ] 入力ファイル名 出力ファイル名¥n"
   "¥t/b¥t¥tヌルの行のとき改行しない¥n"
   "¥t/k¥t¥t禁則処理をしない¥n"
   "¥t/n[num]¥t¥tnum文字ごとに改行する(デフォルト80)¥n"
   "¥t/o¥t¥t基本的に改行をしない¥n"
   "¥t/s¥t¥t文頭がスペースのとき改行しない¥n"
   "¥t/t[num]¥t¥tタブの空白の設定(デフォルトは8)¥n"
   "¥t/?¥t¥tこのメッセージの表示¥n¥n"
   "¥t入力ファイルの文章を指定された字数に改行したものが¥n"
   "¥t出力ファイルになります。¥n"
   "¥t入力ファイルと出力ファイルは重複してはいけません。¥n"
   "¥tスイッチと数字の間に空白を入れないこと。");
   exit(1);
}

void errMes( mes )
char *mes;
{
        puts( mes );
        exit(2);
}

void kaigyou( void )
{
        if( fputc( 10, sakiFile ) == EOF)
                vend = 2;
        moji = 0;
}

void filePut( void )
{
        moji++;
        if( fputc( c1, sakiFile ) == EOF){
                vend = 2;
                return;
        }
        if( zenFlag == 1 ){
                moji++;
                if( fputc( c2, sakiFile ) == EOF){
                        vend = 2;
                        return;
                }
        }
        if( ( c1 == 0x80 ) || ( c1 >= 0xf0 ) ){
                if( fputc( c2, sakiFile ) == EOF){
                        vend = 2;
                        return;
                }
        }
}


int checkKinsoku( void )
{
        unsigned char *po = kin;

        if( kinsokuFlag == 1 ) return 0;
        while( *po != 0){
        if( (*po == c1 ) && ( *(po+1) == c2 ) ) return 1;
        po+=2;
        }
        return 0;
}

int subMain( void )
{
        zenFlag = 0;

        if( ( c1 = fgetc( motoFile ) ) == EOF  ) return 1;
        if( ( ( c1 >= 0x80 ) && ( c1 <= 0x9f) ) || (c1 >= 0xe0) ){
                if( ( c2 = fgetc( motoFile )) == EOF  ) return 1;
                zenFlag = 1;
                if( ( c1 == 0x80 ) || ( c1 >= 0xf0) )
                        zenFlag = 0;
                if( (start == 1) && (c1 == 0x81) && (c2 == 0x40)
                 && (moji != 0) && (spaceFlag == 0) )
                        kaigyou();
        }else
                if( (start == 1) && ( (c1 == 0x9) || (c1 == ' ') )
                && (moji != 0) && (spaceFlag == 0) )
                        kaigyou();

        if(c1 == 10){
                if( (start == 1) && (nullFlag == 0) ){
                        if (moji != 0){
                                kaigyou();
                                kaigyou();
                        } else
                                kaigyou();
                }
                if( ( start == 1) && (nullFlag == 1) )
                        if(moji != 0)
                                kaigyou();
                start = 1;
                return 0;
        } else
                start = 0;

        if( ( moji >= (num-1) ) && ( zenFlag == 1 )
         && ( num != 0 ) ){
                if( checkKinsoku() == 0 || (moji >= num))
                        kaigyou();
                filePut();
                return 0;
        }

        if ( (moji >= num) && (zenFlag == 0) && (num != 0) )
                kaigyou();
        filePut();
        if (c1 == 9)
                moji = ((moji/tab)+1)*tab;
        return 0;
}

void sub( void )
{
        moji = 0;
        start = 1;
        while( vend == 0 ){
                if(subMain() == 1)
                        vend=1;
        }
        if(vend == 2)
                puts("ファイルの書き込みに失敗しました。");
}

void main( argc , argv )
int argc;
char **argv;
{
        int i;

        puts( "X68k Document Adjuster v1.01 Copyright 1992 Tetsuya.Nozaki" );
        if( argc == 1) manual();
        for( i=1; i < argc; i++ ){
        if( (*argv[i] == '-') || (*argv[i] == '/') ) {
                switch (*(argv[i] + 1)){
                case 'b':
                case 'B':
                        nullFlag = 1;
                        break;

                case 's':
                case 'S':
                        spaceFlag = 1;
                        break;

                case 'o':
```

```
		case 'O':
			num = 0;
			break;

		case 'k':
		case 'K':
			kinsokuFlag = 1;
			break;

		case 'n':
		case 'N':
			num = atoi( argv[i] + 2 );
			if (num <= 0) num = 80;
			if (num == 1)
			errMes("文字数が1なのは不当です");
			break;

		case 't':
		case 'T':
		tab = atoi( argv[i] + 2 );
		if (tab <= 0) tab = 8;
		break;

		case '?':
			manual();
			break;
		}
```

```
	}else{
	if( motoName == NULL )
		motoName = argv[i];
	else{
	if(sakiName == NULL)
		sakiName = argv[i];
	else
		errMes("ファイルの指定が多すぎます");
	}
	}
	}
	if( motoName == NULL )
		errMes("ファイルの指定がされていません");
	if( sakiName == NULL )
		errMes("出力先のファイルの指定がされていません");
	if( ( motoFile = fopen( motoName , "r" )) == (FILE*)NULL )
		errMes("ファイルが存在しません");

	if( ( sakiFile = fopen( sakiName , "w" )) == (FILE*)NULL )
		errMes("出力ファイルが作れません");

	sub();

	fclose( motoFile );
	fclose( sakiFile );
}
```

# ぱーてぃハンズ(3)

えー，先月号でキャラクターパターンのリストが掲載されたんですけど，もうパターンは見てもらえましたでしょうか？　今月はこれらのキャラクターを使って，いよいよ殴る蹴る(ううっ，こう書くと集団リンチみたいだ。いかんいかん)のアニメーションを作ります。

さて，よくテレビアニメーションについての解説(っていうのかな)で耳にしているかとは思うのですが，絵を動いたように見せるっていうのは，ある程度以上速いスピードで見せる絵の内容をなめらかに変えてやればいいわけです。で，これの絵の中身がなめらかでないと動きがギクシャクしてしまったり，わけのわからないことになってしまうわけですね。

ところで，映画やテレビアニメっていったい1秒間に何枚くらいの絵を表示しているんでしょう。いろいろ調べてみたところ3～4枚から，30枚！までいろいろあるんだそうですね。

で，これから，このゲームでもキャラクターを，なぐったり，蹴ったりの動きをアニメーションで表現したいわけですが，30枚ってのは論外としても，はたしてBASICでそんなに短い時間に何枚も表示できるもんなんでしょうか。ふと，不安を感じずにはいられないわけです。

## 歩く！　打つ！　飛ぶ！

で，いろいろな絵を表示させてみたところ，BASICではナカワリ(っていうんだそうですね，ある動作からある動作をアニメーションするときに間にはさむ動作の絵のことを)を表示させるのはスピードの面でかなり無理があるみたいです。というのも，中割りや，タメの動作(っていうのかな。野球のバットスイングでいえば，テイクバックみたいなの)をやると，動きがなめらかなのはいいんですが，ジョイスティックのボタンを押してから，キャラクターの動作が非常に緩慢になってしまうのですね。

ということで，今回のこのゲームでは「タメの動作からいきなり打ちの動作にいってしまう！」ってことで解決してしまうことにしました。もちろん，ゲーム中に動きが緩慢なせいでストレスが溜まらないようにという配慮からこうしたのです。しかし，実際にゲームを作るときにはどちらのほうがいいかは，もうそのゲームの味を決めるわけですから好きなほうを選択していいわけです(実際，素早く動くより，リアルに絵が動くほうが好きって人もいるでしょうからね)。ま，とにかく，このへんはゲームの仕様を決めるときにいちばん吟味しなくてはいけない部分でしょうね。

さて，それでは，キャラクターのできる動きについて決めていきましょう。

スプライトのキャラクターパターンを見た人ならもうわかっているとは思いますが，行う動作は「タメ」「打ち」「蹴り」「ジャンプ」「歩き」の5つです。

では実際に動かしてみましょう。

## 動くんだよん

さて，ジョイスティックやボタンの動作に合わせてアニメーションを割り振らなくてはなりませんね。いろいろ，やり方はあるかと思いますが，とりあえず，今回はこんな具合に割り振ってみました。

●ジョイスティック

ニュートラル(5)……立ち

左，右(4，6)……歩き

上(8)……ジャンプ

トリガA……パンチ

トリガB……キック

これをそのままプログラムとして割り振っていけばいいわけです。で，作ってみたものが今月のリストです。

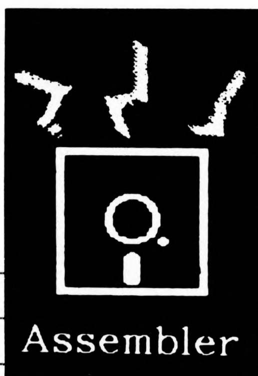
これは先ほど述べたとおり，ボタンが押されたり，ジョイスティックが決まった方向に倒されたときに，キャラクターの絵を切り替えればいいわけです。

ところが，このリストのプログラムを実際にRUNさせてみると，ちょっと動作がおかしいのがわかります。それは，キックやパンチをしたままジョイスティックの左右の動きに反応して歩いてしまうのですね。

ということで次回はいままでのリストを1本にまとめてちゃんと動けるようにしてみましょう。また来月。

リスト

```
1000 screen 0,0,0,0
1001 int x,y,h
1002 x=48:y=128:dh=-4
1010 sp_disp(1)
1020 sp_on(0,1)
1030 PutSprite(0,x,y)
1035 if(stick(1) = 4 and h = 0) then walk(x,y):x=x-4
1036 if(stick(1) = 6 and h = 0) then walk(x,y):x=x+4
1037 if(stick(1) = 8 and h = 0) then h=-8:dh=-8
1040 if(strig(1) = 1 and h = 0) then PutSprite(2,x+4,y)
1042 if(strig(1) = 2 and h = 0) then PutSprite(26,x+4,y)
1044 if(strig(1) = 0 and h = 0) then PutSprite(0,x,y)
1045 if(h<>0) then h=h+dh:PutSprite(6,x,y+h):if(h<-64) then dh=8
1050 goto 1035
8050 func walk(x, y)
8060    int ix,iy
8070    for  ix=0 to 1
8080       iy=2
8090       sp_move(ix+iy*8,x+ix*16,y+iy*16,8 +ix+iy*8)
8110    next
8120 endfunc
9050 func PutSprite(pnum, x, y)
9060    int ix,iy
9070    for  ix=0 to 1
9080       for iy=0 to 2
9090          sp_move(ix+iy*8,x+ix*16,y+iy*16,pnum+ix+iy*8)
9100       next
9110    next
9120 endfunc
```




# Human68k Ver.2.0の機能

Murata Toshiyuki 村田 敏幸

今月から前後編でHuman68k Ver.2.0の拡張機能を解説しましょう。まず前編は，ファイル入出力まわりについてです。ディスクの大容量化にともなう注意点のほか，ファイルシェアリング機能，仮想ドライブ/仮想ディレクトリ機能などを紹介します。

今月と来月の2回で，なんとなく放ったらかしになっていたHuman68k Ver.2.0の拡張機能をやっつけてしまおうと思う。ご存じのように，初期のころはMS-DOS Ver.2.1に毛が生えた程度の代物だったHuman68kも，Ver.2.0になった時点でバックグラウンドプロセスの導入をはじめとする大小の機能強化がなされ，パーソナルコンピュータ用のOSとしてある程度の水準には達した(ことにしておこう)。それにともない，DOSコールもずいぶん拡張/新設されている(表1)。今回は，このうち主として入出力まわりを扱う。なお，個々のDOSコールの呼び出し方などについてはXC Ver.2の『プログラマーズマニュアル』を参照してもらうということで，本稿ではむしろマニュアルでは触れていないような細部にこだわってみたい。

表1 Human68k Ver.2.0で拡張/新設されたDOSコール

| | | |
|---|---|---|
| $FF0F_H$ | DRVCTRL | |
| $FF17_H$ | FATCHK | |
| $FF33_H$ | BREAKCK | |
| $FF3D_H$ | OPEN | |
| $FF44_H$ | IOCTRL | |
| $FF4B_H$ | EXEC | |
| $FF55_H$ | COMMON | (新) |
| $FF58_H$ | MALLOC2 | (新) |
| $FF5A_H$ | MAKETMP | (新) |
| $FF5B_H$ | NEWFILE | (新) |
| $FF5C_H$ | LOCK | (新) |
| $FF5F_H$ | ASSIGN | (新) |
| $FF7C_H$ | (非公開) | (新) |
| $FF7D_H$ | S_MALLOC | (新) |
| $FF7E_H$ | S_MFREE | (新) |
| $FF7F_H$ | S_PROCESS | (新) |
| $FFF3_H$ | DISKRED | |
| $FFF4_H$ | DISKWRT | |
| $FFF5_H$ | INDOSFLG | (新) |
| $FFF6_H$ | SUPER_JSR | (新) |
| $FFF7_H$ | BUS_ERR | (新) |
| $FFF8_H$ | OPEN_PR | (新) |
| $FFF9_H$ | KILL_PR | (新) |
| $FFFA_H$ | GET_PR | (新) |
| $FFFB_H$ | SUSPEND_PR | (新) |
| $FFFC_H$ | SLEEP_PR | (新) |
| $FFFD_H$ | SEND_PR | (新) |
| $FFFE_H$ | TIME_PR | (新) |
| $FFFF_H$ | CHANGE_PR | (新) |

## 大容量デバイスへの対応

たぶん，ユーザーにとって最も嬉しい改良点は容量の大きなディスクがつながるようになったことだろう。Human68k Ver.1.0では1ボリュームとして扱えるディスクの容量は約64Mバイトまでという制限があり，それ以上の容量をもつハードディスクは複数のパーティションに分割して使うよりなかった(しかも，同時に使えるのは1パーティションのみ)。Ver.2.0ではこの上限が大幅に拡大され，SCSIデバイスドライバが提供されたのとあいまって，光磁気ディスクや数百Mバイト級(なんならGバイト級)のハードディスクも利用できるようになった。

大容量デバイスへの対応は，ディスクの管理方法を自然に拡張することで行われた。Ver.1.0では1セクタのバイト数，1クラスタあたりのセクタ数が，それぞれ，1024バイト/セクタ，1セクタ/クラスタに固定されていたのが可変になったのだ。ただし，ともに2の整数乗の値でなければならず，1セクタの大きさは32バイト以上32Kバイト以下，1クラスタあたりのセクタ数は1以上256以下という制限はある[1]。また，いまのところ純正デバイスドライバではこれまでどおり1セクタ=1024バイトのまま，ディスク容量に応じて1クラスタあたりのセクタ数を増やすことで大容量ディスクに対応しているようだ。大雑把にいって，64Mバイトまでは1セクタ/クラスタ，128Mバイトまでは2セクタ/クラスタ，以下，容量が倍になるにつれ，1クラスタあたりのセクタ数も倍々になっていく。

64M，128M，256Mバイトというあたりに切り替え点があるのは，総クラスタ数が約65,000個までに

1) この数字はDPBの構造そのほかから類推される論理的な限界であり，実際に試してみたわけでも，Human68kを解析したわけでもないことを断っておく。きっとHuman68k自体は上限まで対応しているのだろうが，個々のデバイスドライバやフォーマットユーティリティがどこまで対応しているかは不明だ。

制限されていることによる(Ver.1.0, Ver.2.0とも)。これはクラスタの使用状況を管理するFAT(File Allocation Table)が16ビット長までのクラスタ番号にしか対応していないためだ。なお，理論上は，FATを拡張してクラスタ番号の上限を増やすことでも大容量化はできる。しかし，それにつれてFAT領域に必要なディスク容量が急激に増えるので，あまり現実的ではない。12ビットFATは最大でも6Kバイトにしかならないが，16ビットFATなら128Kバイト，仮にもう4ビット増やして20ビットFATにしたとすると，実に2.5Mバイトが必要になる。

さて，いくら大容量ディスクが使えるようになったからといって，それを1ボリュームで使うのはあまり得策ではない。Human68kはファイルをクラスタ単位で管理しており，各ファイルは最低でも1クラスタを占める[2)]。ファイルサイズが大きくなって1クラスタに収まらなくなったら，もう1クラスタ，というように空きクラスタが割り当てられていく[3)]。クラスタサイズに満たない端数にも1クラスタが割かれることに注意しよう。ファイルの最後のクラスタには使われることのない無駄な部分があるわけだ。比較的小さなファイルが多数ある場合，クラスタサイズが大きいと，それだけ無駄になるディスクスペースも増える。したがって，大容量ディスクも，1クラスタ＝1セクタ＝1024バイトになる64Mバイト以下の複数パーティションに分割して使ったほうが，ディスクの使用効率の点では好ましい[4)]。また，複数パーティションに分割することには，(物理的なディスクの破壊はともかく)論理的にFATが壊れるといった事故が起きた場合にドライブ全部がもっていかれてしまうのを回避するという意味もある[5)]。

ディスクの大容量化がプログラムにどう影響するかだが，ふつうにファイルの入出力を行う程度なら，なにも変わらないといってよい。ただ，ディレクトリのソートやパックを行うプログラムなど，セクタ単位でディスクを直接読み書きするプログラムを作る場合は，Ver.1.0時代の1クラスタ＝1セクタ＝1024バイトという仮定が通用しないことを頭に入れておこう。DOSコールgetdpbでDPB (Drive Parameter Block)と呼ばれるドライブについての情報を取得し，必要なデータを入手しなければならない。DPBの構造は『プログラマーズマニュアル』に載っている[6)]が，縁起ものだから表2にもまとめておいた。また，表3にメディアバイトの一覧も用意した。表3にはメーカー純正デバイスドライバでサポートされている(将来のための予約も含む)もののみを記した。サードパーティ製のデバイスドライバやフリーソフトウェアではMS-DOSからの類推でほかのメディアバイトを使用している場合もある。

セクタ入出力まわりではDOSコールも一部拡張されている。Human68k Ver.1.0では16ビットに収まったセクタ番号の最大値がVer.2.0ではそれ以上になりうるので，セクタ番号を直接扱うDOSコールchkfat，diskred，diskwrtで32ビットのセクタ番号がサポートされた。互換性確保のためにこれまでどおり16ビットのセクタ番号を使う呼び出し方法も残されているが，これから作成するプログラムは32ビットのセクタ番号を使うようにしたほうがよいだろう。あるいは，Human68kのバージョンをチェックして2つの呼び出し方法を使い分けることも可能だ(あんまり，報われないような気がしないでもないが)。

話は戻って，ちょっとDPBの話をしておきたい。マイナーな話になるのだが，Ver.1.0とVer.2.0ではDPBまわりに微妙な違いがあり，互換性を損ねている。まず，DPB中のカレントディレクトリ。Ver.1.0ではパスの区切りとして“/”を使っていた[7)]のに対し，Ver.2.0ではほかとの統一性からか“¥”になった。これに加えて，Human68k内部でのDPBの構造が変更されている。Ver.1.0ではHuman68k内部でもDOSコールgetdpbが返す構造そのままの形でDPBを保持していたのだが，Ver.2.0はそうではない。Ver.2.0のgetdpbは内部形式のDPBをVer.1.0と同じ構造に加工して返すことで互換性を保っているわけだ。しかし，DPB中には“次のDPBへのポインタ”がある。このポインタはHuman68k内部のDPBをじかに指しており，当然，その先にあるのは内部形式のDPBだ。このため，Ver.1.0では有効だった“ポインタをたどってDPBを順に参照する”という方法がVer.2.0では使えなくなった。

ちなみに，Ver.2.0の内部DPBは表4のような構造をしている。未公開情報なのであくまで参考のつ

**表2　DPBの構造**

| | | |
|---|---|---|
| $00_H$ | 1b | ドライブ番号 (0=A:, 1=B:, ...) |
| $01_H$ | 1b | ドライバ内のユニット番号 |
| $02_H$ | 1w | 1セクタあたりのバイト数 |
| $04_H$ | 1b | 1クラスタあたりのセクタ数－1 |
| $05_H$ | 1b | クラスタ→セクタのシフト数 |
| $06_H$ | 1w | FAT領域先頭セクタ番号 |
| $08_H$ | 1b | FAT領域の個数 |
| $09_H$ | 1b | 1個のFAT領域のセクタ数 |
| $0A_H$ | 1w | ルートディレクトリに入るファイル数 |
| $0C_H$ | 1w | データ領域先頭セクタ番号 |
| $0E_H$ | 1w | 総クラスタ数＋1 |
| $10_H$ | 1w | ルートディレクトリ先頭セクタ番号 |
| $12_H$ | 1l | デバイスドライバへのポインタ |
| $16_H$ | 1b | メディアバイト |
| $17_H$ | 1b | 書き込みが行われたかどうかのフラグ (－1でアクセスなし) |
| $18_H$ | 1l | 次のDPBへのポインタ(－1なら末尾) |
| $1C_H$ | 1w | カレントディレクトリのクラスタ番号 |
| $1E_H$ | 64b | カレントディレクトリ名 |

2)　Human68kではファイルを新規作成したときに，無条件に1クラスタを確保するので，0バイトのファイルにも1クラスタが割かれる。

3)　Human68kは空きクラスタを使い切ったときに次のクラスタを確保する。このため，ちょうどクラスタサイズの倍数のファイルを作ると，1クラスタ余計に確保されてしまう。

4)　同じ理由で，緻密なディスク管理を行うために，1クラスタのバイト数を標準以上に小さくした(512バイトとか256バイトとか)RAMディスクドライバなんていうものも考えられる（どこかに実在するかもしれない)。

5)　唐突ながら，本体とは独立して電源が切れるようになっていない限り，僕がハードディスク内蔵マシンを買うことはないだろう。

6)『プログラマーズマニュアル』にはgetdpbの項と，デバイスドライバの項の2カ所にDPBの構造が記されているが，前者には$05_H$バイト目の説明に誤りがある（XC Ver.2.0のマニュアルで確認。Ver.2.1もきっと同じだろう)。

7)　ご存じとは思うが，Human68kではパスの区切りとして“¥”と“/”の両方を受け付ける。

**表3　メディアバイト**

| | |
|---|---|
| $F4_H$ | DAT |
| $F5_H$ | CD-ROM |
| $F6_H$ | MO |
| $F7_H$ | SCSI-HD |
| $F8_H$ | HD |
| $F9_H$ | RAM-DISK |
| $FE_H$ | 2HD |

もりで見てほしい。カレントディレクトリ名など，一部の足りない情報は別の領域に格納されており，代わって内部で使用する情報が加わっている。ここでは“空きクラスタの検索開始位置”がちょっと目新しい。

Human68kは，ファイルを新規作成したり書き込み中のファイルが大きくなったりして新たなクラスタを確保する必要が生じた場合にFATを検索して空きクラスタを探すわけだが，Ver.1.0ではこの検索を常にFAT先頭から行い，最もクラスタ番号の小さな空きクラスタを確保するようになっていた。対して，Ver.2.0では各ドライブごとに前回どこまで検索したかを内部DPBに保持しておいて，次回の空きクラスタ検索はその位置から始める。その間にファイルが削除されて前のほうに空きができたとしても，FATの検索が末尾に達するか，再起動するか，あるいは，DOSコールdrvctrlのサブファンクション10(Ver.2.0での拡張)を使って強制的にFATの検索位置をリセットするまで，そこは使われない。

この様子を表5に示した。表5は，Human68k Ver.2.0でG-RAM上にとったRAMディスクに，別ドライブから200Kバイトの同じファイルを続けて何度もCOPYコマンドでコピーし，そのファイルがどのセクタに収まったかを調べた結果だ。同じファイルを上書きする形でコピーしているにもかかわらず，ファイルの位置がどんどんずれていくのがわかるだろう。途中，FATの検索位置が最終クラスタに達した時点で，折り返してふたたび先頭から探し始める様子も確認できる。

想像するに，FAT検索アルゴリズムが変更されたのは空きクラスタの検索時間を短縮するためだろう。Ver.1.0のやり方では，すでにびっしり埋まっている可能性の高いFAT前半部を空きクラスタを探すたびに走査する必要があるが，Ver.2.0方式なら一度ですむ。FAT領域は最大で128Kバイトになるから，仮に前半分が埋まっているとすると，64Kバイト分のディスク読み込みとメモリテストが省ける計算だ。細かなことをいえば，Human68kは内部にディスクキャッシュをもっているので，Ver.1.0でも実際のディスク読み込み量はもっと少ないだろう。しかし，FATすべてがキャッシュ上にあったとしても，メモリテストの時間がなくなればかなりの効率アップは期待できる。また，Ver.2.0方式ではFAT領域のうちいま注目している1セクタのみがメモリ上にあればよいわけであり，キャッシュもより有効に働くと考えられる。

ところで，表5を注意してみると，1クラスタだけ常に重複していることに気づく。このことから，内部DPBに保持されるのは最後に確保したクラスタ番号そのものだということがわかる。連続してクラスタを確保する場合のことを考えると，最後に確保したクラスタ番号＋1を保持するほうがテストが1回省略できてわずかに効率がいいんじゃないか，と，つい思ってしまった。

効率といえば，ファイルの入出力速度には，ファイルが連続したクラスタに格納されているかどうかがかなり影響する。ファイルが飛び飛びのクラスタに分断して格納されていると，ファイル全部を読み書きするまでのあいだにドライブのヘッドが物理的に移動する回数/距離が大きくなり，それだけ入出力が遅くなる。とくにファイルの作成/削除を繰り返したディスクではファイルが分断される確率が高くなり，その度合が進むと，人間にも速度低下がわかるほどになる。ディスクは使い込むにつれ，アクセス速度が劣化していくのだ。

ここで，FATの検索アルゴリズムの変更は，ファイルが分断される率，ひいてはディスクアクセス速度の劣化しやすさも変える，と考えられる。実際にどう変わるかは，ファイルの作成/削除の頻度やディスク容量なども関係してくるだろうから一概にいえるものではない。よくなるか悪くなるかすらはっきりとはわからない。だが，プログラムの開発過程など，一時期にファイルの作成/削除を何度も繰り返す場合を想定すると，Ver.1.0方式はディスクの比較的狭い領域を引っ掻き回すがために部分的なファイルの“もつれ”は起きやすそうだ。その代わり，デ

**表4　内部DPBの構造(参考)**

| | | |
|---|---|---|
| $00_H$ | 1b | ドライブ番号（0=A:, 1=B:, ……） |
| $01_H$ | 1b | ドライバ内のユニット番号 |
| $02_H$ | 1l | デバイスドライバへのポインタ |
| $06_H$ | 1l | 次のDPBへのポインタ（−1なら末尾） |
| $0A_H$ | 1w | 1セクタあたりのバイト数 |
| $0C_H$ | 1b | 1クラスタあたりのセクタ数−1 |
| $0D_H$ | 1b | クラスタ→セクタのシフト数 |
| $0E_H$ | 1w | FAT領域先頭セクタ番号 |
| $10_H$ | 1b | FAT領域の個数 |
| $11_H$ | 1b | 1個のFAT領域のセクタ数 |
| $12_H$ | 1w | ルートディレクトリに入るファイル数 |
| $14_H$ | 1w | データ領域先頭セクタ番号 |
| $16_H$ | 1w | 総クラスタ数＋1 |
| $18_H$ | 1w | ルートディレクトリの先頭セクタ番号 |
| $1A_H$ | 1b | メディアバイト |
| $1B_H$ | 1b | セクタ→バイトのシフト数 |
| $1C_H$ | 1w | FAT検索開始位置 |

**表5　FAT検索位置移動の様子　その1**

| | |
|---|---|
| 1回目のコピー後 | $0004_H$-$00CC_H$ |
| 2回目のコピー後 | $00CC_H$-$0194_H$ |
| 3回目のコピー後 | $0194_H$-$01FF_H$ $0004_H$-$0060_H$ |
| 4回目のコピー後 | $0060_H$-$0128_H$ |
| 5回目のコピー後 | $0128_H$-$01F0_H$ |

ィスクの使用中の部分と未使用の部分がはっきり分かれるので，ある程度ディスクを使い込んだあとでも，ディスク後半部には連続した空きクラスタが残っていることが期待できる。Ver.2.0方式のほうは部分的なファイルのもつれは起きにくそうだが，ディスクの空き領域がいくつかの中小ブロックに区切られる率は高くなる。残り容量が少なくなると急激にアクセス速度の低下を招くかもしれない。

と，ここまで書いた時点で，ふと思いついたことがあったのでまた実験してみた。表6は先ほどと同様の実験を，COPYコマンドではなく手元にあったファイルコピーユーティリティで行った結果だ。見てのとおり，2回目のコピー以降は先頭1クラスタだけが動かずに孤立してしまっている。このコピーユーティリティが特別変な作りになっているというわけではない。種を明かせば，単に“Cで書かれている”だけなのだ。こと，この問題に関する限り，Human68k Ver.2.0のFAT検索アルゴリズムは，XCのライブラリ関数open()との相性が悪い。

もともとopen()はUNIXのシステムコールであり，UNIX上のCでは単なるOSインタフェイス(XCでいうDOSLIB相当)として用意されている。CとUNIXは切っても切れない関係なので，ほかのOS上のCでもUNIXのシステムコールをエミュレートするライブラリ関数としてopen()がサポートされる場合が多い。XCも一応そうだ。で，UNIXの仕様では，open()でファイルを新規作成する際には，

書き込みモードでオープン
存在するならサイズを0に切り詰める
ファイルがなければ新規作成

という指示を組み合わせて与えることになっている。これをHuman68kのDOSコールでエミュレートすると，こうなる。

1) DOSコールopenで書き込みモードでオープンする
2) オープンできたら，DOSコールwriteでファイルサイズを0に切り詰める[8)]
3) オープンできなかったら(ファイルが存在しなかったら)DOSコールcreateで新規作成する

ファイルがすでに存在するかどうかで，発行されるDOSコールが異なるところがポイントだ。ファイルが書き込み可能な状態でオープンされ，ファイルサイズが0になるという点では，createを使っても，openとwriteを組み合わせても，違いはない。しかし，createはあらかじめ旧ファイルを削除するのに対して，writeでファイルサイズを0に切り詰めたときには先頭1クラスタが解放されずにそのまま残る。それが表6のような1クラスタ生き別れ現象を引き起こすのだ。ちなみに，ファイルの作成にcreat()やfopen()などの関数を使っても，結局は内部でopen()が呼び出される(XCの場合)ので状況は変わらない。

念のためだが，ここで問題にしているのは相性であって，べつにHuman68k Ver.2.0のFAT検索アルゴリズムが悪いわけでも，XCのopen()関数が悪いわけでもない。強いて悪者を探せば，Human68kが空きクラスタを確保するタイミング，だ。ファイルを作成するときに1クラスタ確保し，1クラスタ使い切ったら(さらに書き込みが行われるかどうかは考えずに)その場でもう1クラスタ確保する，というように先走るのが諸悪の根源。実際にデータが出力されたときに，それまで使っていたクラスタがすでにいっぱいだったら新たに確保する，というようになっていれば，0バイトのファイルに1クラスタ消費するような無駄も含めて，多くの問題は解決するだろう。

## ファイルシェアリング

話題は変わって，ファイルシェアリング(file sharing)機能をみてみよう。これは，並行して動作する複数のプログラム，あるいは，ネットワーク下の複数のユーザーが，同じファイルを共有できるようにするための仕組みだ。

書き込み中のファイルをまた別にオープンして交互に書き込みを行えば，ほぼ確実にファイル内容はぐちゃぐちゃになる。最悪の場合，FATやディレクトリに矛盾を生じて，

ディスクの管理領域が破壊されています

というエラーメッセージを拝むことにもなりかねない。ところが，複数のプログラムが並行して走っていたり，ネットワークで外部記憶装置を共有するほかのユーザーがいたりすると，互いにそれとは知らずに同じファイルに書き込む可能性が出てくる。ファイルシェアリングをサポートしたということは，こんな事故を防ぐために，排他的にファイルアクセスできることを保証する(＝同一ファイルに対するよそからのアクセスを禁止できる)安全装置が備わった，ということだ。この意味で，“ファイルアクセスの排他制御”という言い回しが，ファイルシェアリングと同じ文脈で使われる。

8) writeで書き込みバイト数に0を指定すると，その時点でのファイルポインタ（次の読み書き位置）以降が切り捨てられる。

**表6 FAT検索位置移動の様子 その2**

| | |
|---|---|
| 1回目のコピー後 | $0004_H$-$00CC_H$ |
| 2回目のコピー後 | $0004_H$-$0004_H$ $00CC_H$-$0193_H$ |
| 3回目のコピー後 | $0004_H$-$0004_H$ $0193_H$-$01FF_H$ $0005_H$-$005F_H$ |
| 4回目のコピー後 | $0004_H$-$0004_H$ $005F_H$-$0126_H$ |
| 5回目のコピー後 | $0004_H$-$0004_H$ $0126_H$-$01ED_H$ |

Human68kにおけるファイルアクセスの排他制御には，ファイル単位のものと，ファイルの部分領域単位のものがある。ファイル単位の排他制御はDOSコールopenのオープンモードを拡張するという形で実装された。オープンモードの第4～6ビットでシェアリングモードを指定することで，そのファイルをそれ以上オープンすることを禁じたり許可したりできる(図1)。その効果はオープンを禁止/許可したプログラム自身にもおよぶ。つまり，オープンを禁じたファイルは，自分でもオープンできなくなる。

部分領域の排他制御は新設されたDOSコールlockで行う。lockは，ファイルのバイト位置と長さで指定した任意の領域をロック(よそからのアクセスを禁止する)したり，ロック解除したりする。データベースファイルなどのように，ファイルが複数のレコードから構成され，各レコードが独立して参照/更新される可能性のある場合に有用な機能だ。

表7　有効なオープンモードの組み合わせ

| | $00_H$ | $01_H$ | $02_H$ | $10_H$ | $11_H$ | $12_H$ | $20_H$ | $21_H$ | $22_H$ | $30_H$ | $31_H$ | $32_H$ | $40_H$ | $41_H$ | $42_H$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $00_H$ | ○ | × | × | × | × | × | × | × | × | × | × | × | × | × | × |
| $01_H$ | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × |
| $02_H$ | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × |
| $10_H$ | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × |
| $11_H$ | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × |
| $12_H$ | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × |
| $20_H$ | × | × | × | × | × | × | ○ | × | × | × | × | × | ○ | × | × |
| $21_H$ | × | × | × | × | × | × | × | × | × | ○ | × | × | ○ | × | × |
| $22_H$ | × | × | × | × | × | × | × | × | × | × | × | × | ○ | × | × |
| $30_H$ | × | × | × | × | × | × | × | ○ | × | × | × | × | × | ○ | × |
| $31_H$ | × | × | × | × | × | × | × | × | × | × | ○ | × | × | ○ | × |
| $32_H$ | × | × | × | × | × | × | × | × | × | × | × | × | × | ○ | × |
| $40_H$ | × | × | × | × | × | × | ○ | ○ | ○ | × | × | × | ○ | ○ | ○ |
| $41_H$ | × | × | × | × | × | × | × | × | × | ○ | ○ | ○ | ○ | ○ | ○ |
| $42_H$ | × | × | × | × | × | × | × | × | × | × | × | × | ○ | ○ | ○ |

さて，Human68kにおけるファイルアクセスの排他制御機能はCONFIG.SYSのSHARE行を設定して初めて機能する。SHAREを設定しているユーザーは少ないようだが，積極的に利用するつもりはなくても，とりあえず設定だけはしておいたほうがよい。SHAREが設定されていないと，ファイルオープン時にシェアリングモードを指定したり，部分領域をロックしようとしてもエラーが返るだけなので，気を利かせて排他制御を行うプログラムが困る。また，排他制御をとくに意識していないプログラムを実行する際にも，ある種の事故を防ぐ効果がある。

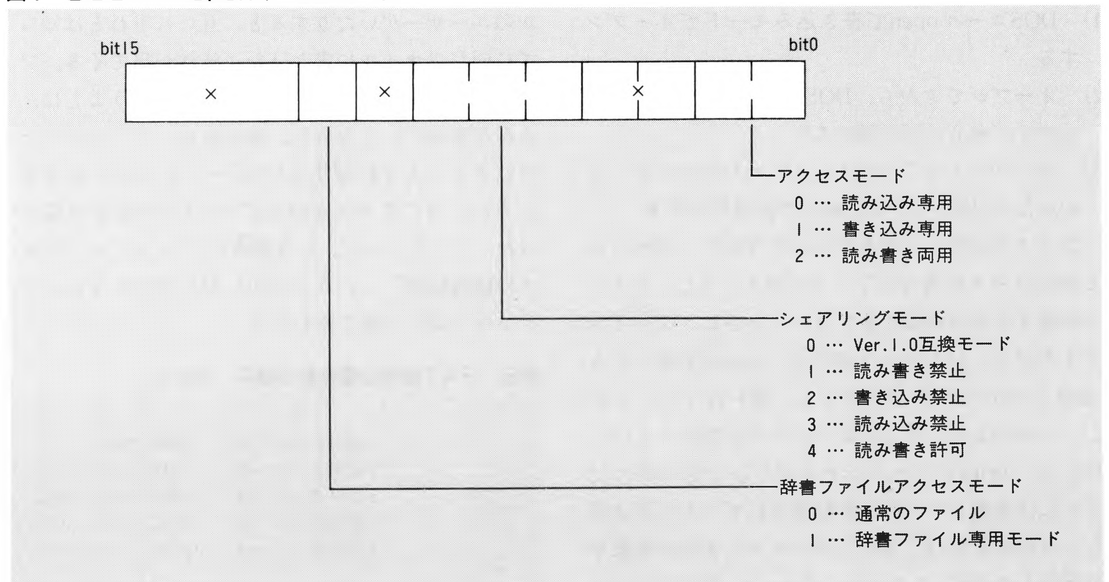
表7は，SHAREが設定された状態で同じファイルを複数オープンする際に指定可能なオープンモードの組み合わせをまとめたものだ。たとえば，オープンモード$30_H$でオープンしたファイルは，それ以降，モード$21_H$，$41_H$のみでオープンできる，というように読む(対角線を軸に対称だから，読む向きは縦でも横でもかまわない)。これをみると，とくにシェアリングモードを指定しなくても(＝互換モードを指定しても)，自動的に排他制御は行われることがわかる。ふつうの読み込み専用モードでオープンしたファイルは以後書き込みモードではオープンできず，また，書き込みモードでオープンしたファイルはいっさいオープンできなくなる。SHAREが設定してあるかどうかで挙動が変わってしまうのにどこが互換モードなんだ，という話もあるが，おかげでファイルを誤って破壊する事故はかなり防げる。

さらに，SHAREを設定しておくと，アクセス中のファイルの削除や属性変更が禁止されるという効用もある。端的な例として，

DEL GHOST >GHOST

を挙げよう。SHAREが設定されていれば，オープン

図1　DOSコールopenオープンモード



中のファイルを削除する形になるので，排他制御が働き，GHOSTの削除は行われない。しかし，設定されていないと，このコマンドは次のように動作して，FATとディレクトリが矛盾した状態を作り出す。

1)　リダイレクト先ファイルGHOSTを作成する。ディレクトリ上には仮にファイルサイズ0でGHOSTについてのファイル情報が書き込まれる。併せて，Human68kの仕様上，1クラスタが確保され，GHOSTに割り付けられる
2)　GHOSTを削除する。1)で確保したクラスタは解放される
3)　リダイレクト先GHOSTをクローズする。このときGHOSTのファイル情報が再度ディレクトリに書き込まれる
4)　結果として，ディレクトリ上に空きクラスタを指す幽霊ファイルの残骸が残る

幽霊ファイルは，すぐに削除してしまえば害はない。エラーは出るが，その時点でディレクトリ上のゴミは消えるので，FATとディレクトリの不整合は解消される。しかし，幽霊というだけあって，放っておくと結構怖い。GHOSTをこのままにしておき，別のファイルFOOを作成すると，FOOはたぶんさっきGHOSTに一時的に割り付けられたクラスタに収まるだろう。それからGHOSTを削除すると，FOOに割り当てられたクラスタが解放され，今度はFOOが幽霊になる。

あまりにわざとらしい例だと思ったかもしれない。確かに，DELは出力をリダイレクトしても意味がないから，ふつうこんなことはしない。とはいえ，

　　DEL ＊.O

という馴染みのコマンドも，ちょっと慌ててSHIFTキーを放すのが遅れれば，

　　DEL ＊>O

に化けるのだ。くどいようだが，SHAREを設定していないようなら，いますぐCONFIG.SYSを書き換えることを勧めたい。

SHAREは2つの引数をとり，第1引数で排他制御を行うファイル数，第2引数で1本のファイルあたりのロック領域数を指定する。ファイル数はFILESで設定したオープン可能な最大ファイル数と同程度にしないと意味がない。というのは，異なるファイルをオープンするごとに排他制御用の情報格納領域が消費され，足りなくなるとファイルのオープン自体できなくなってしまうからだ。事実上，SHAREの第1引数により同時にオープンできるファイルの最大数も制限される[9]。第2引数のほうはそれほど大きな値にする必要はないだろう。ネットワーク環境下なら別だが，いまのところDOSコールlockは滅多に使われない。ただ，同一のファイルを同時に複数オープンする(もちろん，許されるオープンモードで)と排他制御情報は共用され，オープンごとにロック領域が消費されるようだ。SHAREの第2引数によって，“同じファイルを同時にオープンできる回数”が制限される。メモリにゆとりがあれば，やはりFILESに近い値にしておいたほうがいいかもしれない。ちなみに，排他制御情報用のワークには，ファイル1本あたり，

　　ロック領域数×12＋92バイト

のメモリが使用される。

さて，SHAREを設定してもらったばかりで申し訳ないのだが，前述のように，DOSコールopenの互換モードが呼び名のわりには互換性がないので，SHAREを設定すると動かなくなるプログラムというものも考えられる。たとえば，これまでは書き込み位置が重ならなければ，同じファイルを複数オープンして別々にデータを書き込むこともできた。が，SHAREを設定してしまうとシェアリングモードをきちんと指定しない限り，2度目以降のオープンは許されない。また，ふつうなら読み書き両用モードを使うところを，読み込みモードと書き込みモードで2度オープンするような作りのプログラムもうまくない。実際，自作のプログラムで似たような問題が見つかったことがある。そのプログラムはファイルコピーユーティリティの類(さっきの実験に使っ

**表8　DOSコールioctrlサブファンクション追加分(参考)**

```
サブファンクション9
move.w    #ドライブ番号, -(sp)
move.w    #9, -(sp)
DOS       _IOCTRL
addq.l    #4, sp
サブファンクション10
move.w    #ファイルハンドル, -(sp)
move.w    #10, -(sp)
DOS       _IOCTRL
addq.l    #4, sp
サブファンクション11
move.w    #リトライ間隔 (1/100単位), -(sp)
move.w    #リトライ回数, -(sp)
move.w    #11, -(sp)
DOS       _IOCTRL
addq.l    #6,sp
サブファンクション12
pea.l     バッファアドレス
move.w    #機能番号, -(sp)
move.w    #ファイルハンドル, -(sp)
move.w    #12, -(sp)
DOS       _IOCTRL
addq.l    #8, sp
サブファンクション13
pea.l     バッファアドレス
move.w    #機能番号, -(sp)
move.w    #ドライブ番号, -(sp)
move.w    #13, -(sp)
DOS       _IOCTRL
addq.l    #8, sp
```

9)　あくまでも“ファイル”をオープンする場合。キャラクタデバイスの場合はオープンしても排他制御情報は作成されないので，SHAREの第1引数を超えた回数のオープンができる。

たものではない)だったのだが，SHAREを設定して以来，ときおりコピーを行わずにエラー終了するようになってしまった。原因はコピー先がファイルかデバイスかを調べる小さなサブルーチンにあった。そのサブルーチンは引数で指定されたファイルを読み込みモードでオープンし，オープンできたらDOSコールioctrlで装置情報を取得して，その第7ビットを調べてccrに返す，という処理を行っていたのだが，クローズするのを忘れていたのだ。そのため，コピー先ファイルがすでに存在すると，読み込みモードでオープン中のファイルを書き込みモードでオープンしなおすことになり，エラーを引き起こしていた。

このような互換性上の障害も，SHAREを設定しない理由にはならないだろう。むしろ，プログラム側が対応するべき問題だ。読者もSHAREを設定した環境で手元のプログラムを走らせてみて，不都合がないかどうか確認してみてほしいと思う。

ファイルシェアリングの話題の最後に，関連して拡張/新設されたopen，lock以外のDOSコールについても触れておく。

ファイルシェアリングというよりはネットワーク関係になってしまうが，DOSコールioctrlのサブファンクションはいくつか増えた。『プログラマーズマニュアル』には載っていないから，未公開機能ということになる。しかし，Cのライブラリには対応する関数が追加されていて，『Cライブラリマニュアル』にもしっかり載っているので，『プログラマーズマニュアル』には載せ忘れたのだろう，と善意に解釈して代わりに解説しておく。表8も適宜参照してもらいたい。

サブファンクション9と10は，それぞれ，ドライブ，ファイルがリモートかローカルかを返す。ここで，リモートとはよそにあってネットワークを介してつながっているデバイス，ローカルはX68000に直接つながっているデバイスのことをいう。結果はd0.lの第12ビットに返り，1ならリモート，0ならローカルだ。第12ビットというのがいかにも変則的だが，実はサブファンクション0で返る装置情報と同じものがd0.lに返ってきている。実際，サブファンクション0と9の処理ルーチンは共通だ。また，これまではドライブ番号から装置情報を得ることができなかったが，サブファンクション10はその手段を提供してくれる(未公開コールの未公開機能を使うことになるのかな)。装置情報はVer.1.0に比べ，

| | | |
|---|---|---|
| ビット6 | 特殊IOCTRL可能 | |
| ビット12 | 0ならローカル | |
| | 1ならリモート | |
| ビット13 | 特殊ブロックデバイス | |

が拡張されているようだ。特殊IOCTRLについては後述する。特殊ブロックデバイスがなにを指すのかはよくわからないが，ビット13が1のときのみビット12が意味をもつことから，ネットワークドライバのことなのだろう。

サブファンクション11はI/Oリトライ間隔と回数を指定する，とある。これはネットワーク下で共有違反などの理由によりアクセスがうまくいかなかったときに，適当な間隔をおいて何度か再リクエストする，その間隔と回数を指定するものらしい。が，ここで設定したI/Oリトライ間隔，回数はDOSコールindosflgで参照できるところをみると，一般のディスクのI/Oリトライ時にも使われうるのかもしれない(つまり， デバイスドライバが自らリトライ間隔と回数をDOSコールで取得して，それだけリトライを繰り返す)。

サブファンクション12と13はデバイスドライバの特殊IOCTRLなるものを行う。早い話がデバイスドライバとの双方向通信だ。ある機能番号を送ったら，それなりのデータを返すようにデバイスドライバを作っておいて，その取り決めにしたがって通信する。ネットワークドライバの制御用に設けられたんじゃないかといい加減に想像しているのだが(全然違うかもしれない)，一般のデバイスドライバが使ってもとくに問題はないだろう。使い方によっては，いろいろ遊べそうだ。デバイスドライバを使ってファンクションコールまがいのものを実現することも可能だろう(遅いだろうけど)。

なお，特殊IOCTRLを行うためには，そのデバイスにIOCTRL可能，特殊IOCTRL可能というデバイス属性を与えておく必要がある。IOCTRL可能にするためには『プログラマーズマニュアル』にもあるように，デバイス属性の第14ビットを立てる。特殊IOCTRL可能にするためには，デバイス属性の第11ビットを立てておけばよいようだ(装置情報との違いに注意)。また，特殊IOCTRL時にはデバイスドライバに対してコマンドコード13でI/Oリクエストが行われ，その際，

```
14(a5).w  機能番号
18(a5).l  バッファアドレス
```

が渡される。この機能を利用したデバイスドライバを作る際の参考にしてほしい。

ところで，サブファンクション12と13に対応するXCの関数はIOCTRLFDCTL()とIOCTRLDVCTL()だ。なのに，IOCTRLFDCTL()がサブファンクション13を，IOCTRLDVCTL()がサブファンクション12を呼び出しており，名前と機能が一致しない。どうもライブラリとHuman68kのどちらかのバグらしい。ライブラリが間違っていると考えるのが自

然だが，ioctrlではドライブ番号とファイルハンドルのどちらにも共通の機能がある場合，ドライブ番号，ファイルハンドルの順にサブファンクション番号が並んでいるのに，サブファンクション12と13だけはこの関係が逆になっているのは怪しい。

バグといえば，サブファンクション13は特殊IOCTRL可能なデバイスかどうかをチェックするのに対して，なぜかサブファンクション12ではチェックがない。誤って対応していないデバイスにこの機能を使おうとすると，最悪バスエラーの類のエラーが発生する可能性がある。事前にサブファンクション0で装置情報を取得して，アプリケーション側で注意しないと危険だ。

さて，新設されたDOSコールnewfileとmaketmpもファイルシェアリングと無縁ではなさそうだ。どちらもcreateと同じ呼び出し方をし，ファイルを作成するという点で機能も似ている。が，newfileは"新規"作成専用で，同名のファイルがすでに存在する場合はエラーを返す[10]。また，maketmpのほうは，

temp????.$$$

のようにファイル名の中に"？"を埋め込んでおくと，そこに適当な数字をはめ込んでユニークなファイル名を作り，その名前のファイルを作成する。作成したファイル名は引数として渡したファイル名格納領域でそのまま返され，ふたたび同じファイル名(を置いたメモリのアドレス)を渡せばまた別の名前のファイルができる。コール名のとおりテンポラリファイル作成用だ。

どちらの機能も既存のDOSコールを組み合わせれば実現可能だ。あえて，ひとつのDOSコールになっているところに，ファイルアクセスの排他制御の匂いが感じられる。ファイルが存在するかどうか調べ，なければ作成するのでは，偶然，バックグラウンドプロセスが同じことをしようとして混乱する可能性があるが，DOSコール中では基本的にバックグラウンドタスクへ制御が移ることがないので，一連の処理を間違いなく完結させることができるわけだ。

## 仮想ドライブと仮想ディレクトリ

Human68kのファイルシェアリングまわりは，実はMS-DOSからの借り物なのだが，一緒に拝借してきた機能に，仮想ドライブと仮想ディレクトリがある。サブディレクトリを疑似的にドライブに割り当てたり(仮想ドライブ)，逆にドライブをサブディレクトリに割り当てたり(仮想ディレクトリ)する機能

10) このためだけに，エラー番号－90が新設されている。

### ディスクキャッシュ

ディスクキャッシュ(cache)とはディスクアクセスを減らす目的でディスク内容をセクタ(などの基本的な)単位でためておくバッファをいう。キャッシュにデータがあれば(キャッシュにヒットする，という)，実際にはディスクを読む必要がなくなるので入力速度が稼げる。ここで，キャッシュ内容とディスク内容は常に一致していないと困るから，書き込み時にはキャッシュとディスクの両方に同じデータを書き込むことで整合性を保つ工夫が必要だ。また，キャッシュにないセクタを読むときには，キャッシュのうちのどれかを破棄してそこに読み込むことになるが，どれを破棄するかはアクセス頻度なり，最後にアクセスした時間なりを基準に選ぶ。

キャッシュの効果は，ディスクの同じ部分を何度も読む場合にとくに有効だ。逆にいろいろなところをまんべんなく読むような場合には余計な処理をする分遅くなることもある。とはいえ，一般にFATやディレクトリなどの管理領域は頻繁にアクセスされるものであり，ディスクキャッシュを用意する意味は十分にある。

本文でいうHuman68k内部のキャッシュとは，CONFIG.SYSのBUFFERで確保したI/Oバッファを指す。I/Oバッファの第1の目的は，ディスクの入出力単位であるセクタの大きさに満たないデータを読み書きする際に，データを一時的にためておくことにあるわけだが，同時にキャッシュとしても働いている。一般に，キャッシュは単に大きくとればよいというものではなく，あまり極端に大きくしてしまうとキャッシュへの問い合わせに時間がかかるようになり，逆効果にもなりうる。しかし，Human68kのBUFFERの設定値はたかだか99個までなので，ある程度メモリにゆとりがあれば，限界まで確保してしまってもいいだろう。

さて，せっかくBUFFERをたくさん確保しても，キャッシュの効果があまり出ない場合もある。COMMAND.XのDIRコマンドはディスクの使用容量/残り容量を知るためにDOSコールdskfreを発行する。dskfreはFATを全部読んで空きクラスタを数えるため，フロッピーディスクならともかく，ある程度の容量のハードディスクになると，とんでもなく遅い。こんなときこそキャッシュに働いてもらいたいものだが，クラスタ数の多い(必ずしも大容量という意味ではない)ディスクの場合，FATがキャッシュに収まりきらなくなる。仮に収まったとしても，キャッシュがFATに占領され，キャッシュの効果は薄れる。以前にも書いたのだが，気になる人はCOMMAND.Xにパッチを当てて，dskfreの呼び出しを，

```
moveq.l #0, d0
```

で潰すのがよいだろう。

なお，キャッシュ内容は，BREAKキーを押すか，DOSコールfflushを呼び出すことでクリアされる。DOSコールdiskredやdiskwrtはキャッシュを通さずにディスクを直接読み書きするので，とくにdiskwrtを使ったときは忘れずにキャッシュをクリアしておかないと，キャッシュとディスクが不整合を起こすので注意したい。

で，XCのライブラリ関数flushall()は意味もなく(ないと思う)fflushを呼び出す。しかも，flushall()はCで書いたプログラムの終了時に無条件に呼び出される。つまり，Cで書いたプログラムを実行するだけで，キャッシュがクリアされてしまうのだ。当然のように，悪い子はXCのライブラリを書き換えて(そのためのソースだ)fflushの呼び出しを削る。この場合，セクタを直接読み書きするプログラムは自発的にFFLUSH()関数を呼び出す必要がある点に注意すること。

だ。ファンクションコールレベルでは新設されたDOSコールassignが，コマンドレベルではSUBST.Xが，これをサポートする。ファイルシェアリングほどではないにしろ，この機能もあまり使われていないらしいので，用途を紹介するとともに，ユーザーレベルでの使用上の注意点に触れておこう。

仮想ドライブは，深い階層にあるサブディレクトリによくアクセスする場合に便利だ。たとえば，

SUBST Z: A:¥LONG¥PATH¥NAME

により仮想ドライブZ:を定義しておくと，以後，

DIR A:¥LONG¥PATH¥NAME

が，

DIR Z:

で簡略指定できるようになる。また，SX-WINDOW関係なんかでは全角文字を使ったディレクトリ名が用いられることがままあり，COMMAND.X上からアクセスするのが面倒なものだが，ここでも，

SUBST Z: A:¥アクセサリ

といった仮想ドライブ定義が有効だ。さらには，データファイルがルートディレクトリにないと動かない間抜けなプログラム(いまだにそんなプログラムがあるのかどうかは知らないが)を騙すのにも使える。そもそもMS-DOSで仮想ドライブがサポートされるようになったのは，「Wordstarがいつまでたっても階層化ディレクトリに対応しなかったから」だという嘘だか本当だかわからない話も残っている。ほかには，

SUBST Z: A:¥

のようにルートディレクトリを仮想ドライブに割り当てることで，疑似的にドライブ名の別名を作り出すこともできる。

なお，上の例ではZドライブを仮想ドライブとして割り当てたが，そのためには，

1) CONFIG.SYSでLASTDRIVE=Z:に設定してある
2) Z:に実ドライブやほかの仮想ドライブが割り当てられていない

という2条件を満たす必要がある。

対する仮想ディレクトリのほうは，接続されている多数のドライブを自分でわかりやすいよう管理するのに利用する。たとえば，ハードディスクが4パーティションに分かれていて，A:～D:を占めている場合，

MD A:¥APPLI || SUBST A:¥APPLI B:
MD A:¥GAME || SUBST A:¥GAME C:
MD A:¥WORK || SUBST A:¥WORK D:

により，B～DドライブがAドライブ下のサブディレクトリであるかのように扱えるようになる。無機的なドライブ名の代わりに，自分の好きな名前をつけることができるわけだ。あるいは，クラスタサイズを小さくする目的で複数パーティションに分割したハードディスクを，見かけ上1ドライブとして使うための手段ともなる。この調子で全ドライブをAドライブ下に束ねればUNIXごっこもできる。

ここで，仮想ディレクトリ名は実在するディレクトリでなければならないので，あらかじめ空のディレクトリを作成しておくのを忘れてはならない。もちろん，空ではないディレクトリに割り当ててもよいが，そのディレクトリ下のファイルにアクセスする手段は失われてしまう。また，仮想ドライブはいわば“別名”であり，元のディレクトリにもいままでどおりアクセスできるのに対して，仮想ディレクトリに割り当てたドライブ名(上の例ならB:～D:)はドライブとして認識されなくなる点にも注意したい。FORMAT.XやDISKCOPY.Xは引数としてドライブ名を要求するから，仮想ディレクトリに割り当てたドライブはフォーマットやディスクコピーができなくなってしまう。したがって，フロッピーディスクドライブを仮想ディレクトリに割り当てるのはあまり勧められない。

あと，SUBST.Xは結構間抜けなプログラムなので，仮想ディレクトリに割り当てるディレクトリや，仮想ディレクトリ名はフルパスで指定する必要がある。また，/Dオプションで割り当てを解除する場合，ドライブ名を与えなければならないという癖もある。仮想ドライブの場合はともかく，仮想ディレクトリを解除する際にも，対応する実ドライブ名を指定する。この変則性は，SUBST.XのせいというよりもHuman68kの仕様上の問題だ。Human68kは，仮想ディレクトリがどのドライブを指すか，ではなく，ドライブがどの仮想ディレクトリに割り当てられているか，というかたちで仮想ディレクトリの割り当てを管理しているのだった。

さて，仮想ドライブ/仮想ディレクトリはユーザーレベルでは，まあ便利な機能といえるだろうが，プログラミングのうえでは気をつけなければならない問題をなにかと増やしてくれた。

まず，仮想ドライブにより，同じファイルを表す複数のパス名ができてしまうため，2つのパス名が同じファイル/ディレクトリを表しているかどうかの検査は面倒になる。事前に，仮想ドライブを実ディレクトリに展開するという前処理が必要となった。もっとも仮想ドライブがなくても，絶対パスと相対パス，半角英字の大文字/小文字，ファイル名の9文字目以降，パスの区切り文字，漢字を含む2バイト文字，といった要素がいろいろあるため，パス名の比較は決して単純な処理ではない(その複雑さは一度味わってみてほしい)。それに比べれば，仮想ドラ

イブの展開ぐらいどうということはないともいえる。にもかかわらず，Human68k純正コマンドが仮想ドライブを考慮していないのには少々驚かされる。

たとえば，COMMAND.XのCOPYコマンドだ。

COPY A:¥DIR¥FILE A:¥DIR¥FILE

のような露骨な指定はきちんと弾いてくれるが，

SUBST Z: A:¥DIR
COPY A:¥DIR¥FILE Z:¥FILE

なんてやると，転送元と転送先の実体が等しいことに気づかずにZ:¥FILEを新規作成し，その結果，転送元ファイルの内容を破壊する(しかも，幽霊化する)。ただ，幸いなことにCOPYは転送元，転送先の順にファイルをオープンするため，SHAREを設定しておけばZ:¥FILEをオープンしようとしたときに排他制御に引っ掛かってエラーで止まる。

逆に，チェックの甘さが利用できる場合もある。COPYALL.Xは，転送元と転送先のドライブが一致しているとコピーを行わない仕様なので，

COPYALL A:¥DIR¥＊.＊ A:¥DIR

はもちろん，なにも問題のない，

COPYALL A:¥DIR¥＊.＊ A:¥BAKDIR

も受け付けない[11]。この不便な仕様も仮想ドライブを使って，

SUBST Z: A:¥BAKDIR
COPYALL A:¥DIR¥＊.＊ Z:¥

のように指定すれば，すり抜けることができる。HISTORY.Xを組み込んでいるのなら，図2のエイリアスも有効かもしれない。

なお，COPYALL.Xが同一ドライブ上でのコピーをサポートしていないのは，

COPYALL A:¥DIR¥＊.＊ A:¥DIR

を弾く意味もあるが，

COPYALL A:¥DIR¥＊.＊ A:¥DIR¥DIR¥

のような指定が行われたときに，再帰的に処理が繰り返されて止まらなくなる[12]のを防ぐのが主眼らしい。ほかにもやりようはあるだろうに，よりによっていちばん楽な方法，つまり，転送元と転送先のドライブが同一なら再帰的な指定が行われた“可能性がある”のでコピーしない，という手抜きをしているのだ。仮想ドライブでCOPYALL.Xを騙すとこんな指定もできてしまうので気をつけてほしい。

余談ながら，パス名の比較をサボって，なおかつ，2つのパス名が同じファイルを指しているかどうか知る方法がないでもない。許されるかどうかは微妙なところだが，その方法ではDOSコールfilesを使う。とにかく，2つのパス名をfilesにかけて，filesが返してきた情報を比較するのだ。通常使用する後半部分のファイル名やらファイルサイズはもちろん，『プログラマーズマニュアル』では意味不明のシンボルでのみ表され，“Human68k内部で使用”とただし書きのついた前半部もとにかく比較してしまう。すべて一致するようなら，2つのパス名は同じファイルを指していると判断できる。filesは完全なファイル名しか受け付けないので，ルートディレクトリと仮想ディレクトリの比較などには利用できないが，通常のファイルやディレクトリであれば，この方法でパス名の同一性が検査できるはずだ。似たような案としては，DOSコールchkfatを使う方法もある。ファイルがどのドライブのどのセクタを占めているのか調べ，それを比較するのだ。ただ，chkfatは大量のメモリを要求するので，filesほど気楽に使うわけにはいかないだろう。

参考までに，Human68k内部で使用される非公開部分を含む，filesが返す情報の構造を表9に示す。非公開部を参照することを勧めるものではないが，これを見てもらえれば，上に示したパス名比較方法がうまく動くことはわかってもらえるだろう。なお，セクタ番号がロングワードになっていることからも想像できるように，表9はHuman68k Ver.2.0でしか通用しない。非公開部の構造はVer.1.0とは異なっている。『プログラマーズマニュアル』ではいまだにVer.1.0の構造が書かれていたりするが，気にしないことにしよう。

パス名同一性の検査以上に，仮想ドライブ/仮想ディレクトリの存在が影響する処理といえば，やはり，セクタ単位での入出力だ。まず，セクタ単位の入出力時は実ドライブ番号がわからないとできない。そこで，仮想ドライブ/仮想ディレクトリを実ディレクトリ/実ドライブに展開して，元のドライブ名を調べる必要がある。この展開だが，仮想ドライブはともかく，仮想ディレクトリの場合はかなりやっかいだ。仮想ディレクトリから実ドライブを簡単に得る方法はないので，全ドライブについてDOSコール

11) A:¥DIR下のファイルをA:¥DIR¥DIR下にコピーしたあと，コピーしたばかりのA:¥DIR¥DIRをさらにA:¥DIR¥DIR¥DIRにコピーする，というように処理が繰り返される。

12) それにしても，COPYALL.Xはなにかにつけて使用法を表示したがって困る。とくに，/Nや/Tオプションを指定したときに“すでにファイルが更新されていてコピーする必要がなかった”場合(明らかに正常動作，正常終了)にまでヘルプメッセージを出すのはなんのつもりだろう。

**図2　COPYALL.Xをだますエイリアス**

```
COPYDIR SUBST Z: %2 | | COPYALL %1 Z: | | SUBST /D Z:
```

**表9　DOSコールfiles/nfilesが返す情報(参考)**

| | | |
|---|---|---|
| $00_H$ | 1b | 検索するファイル属性 |
| $01_H$ | 1b | 検索するドライブ番号(0=A:, 1=B:, ...) |
| $02_H$ | 1l | 検索中のディレクトリのセクタ番号 |
| $06_H$ | 1w | ディレクトリの残りセクタ数 (ルートの場合のみ有効) |
| $08_H$ | 1w | ディレクトリ上のセクタ先頭からのオフセット (-1は該当ファイルなし) |
| $0A_H$ | 8b | 検索するファイル名 |
| $12_H$ | 3b | 検索する拡張子 |
| $15_H$ | 1b | 見つけたファイルのファイル属性 |
| $16_H$ | 1l | 見つけたファイルのタイムスタンプ |
| $1A_H$ | 1l | 見つけたファイルのファイル長 |
| $1E_H$ | | 見つけたファイルのファイル名 |

13）“—”はMS-DOSではファイル名に使えてHuman68kでは使えない唯一の文字だった。

assignで割り当て情報を取得し，仮想ディレクトリに割り当てられているものがあったら，パス名を比較する，という手順を踏むことになる。仮想ディレクトリに関しては，filesやchkfatを使う手も通用しない。仮想ドライブ用に用意した元のディレクトリの情報しか得られないのだ。

また，仮想ディレクトリに割り当てられた実ドライブは，単にドライブ名が無効になるだけではなく，ドライブ番号のレベルで無効になってしまう。このため，getdpbやdiskred/diskwrtを使うためには，先に展開して得た実ドライブがもし仮想ディレクトリの割り当て先になっていたら，それを解除して，あとで元に戻す必要がある。ここでは，元のパス名が仮想ディレクトリを含んでいたかどうかではなく，展開後の実ドライブに注目することに気をつけよう。割り当て方によっては，仮想ドライブを展開した結果得られた実ドライブが，別の仮想ディレクトリに割り当てられている場合もあるからだ。

## そのほかの機能

些細なところでは，DOSコールbreakckが拡張され，ブレイクチェックを完全に殺すことができるようになった。併せて，COMMAND.XやCONFIG.SYSのBREAKでは“ON”，“OFF”に加えて“KILL”が追加された。ただし，この機能が完全に動作するのは現時点での最新バージョンであるVer.2.03からだ。Ver.2.02までは，BREAK KILLに設定してあっても，CTRL＋S（またはSHIFT＋BREAK）で表示を一時停止した状態だとBREAKキーが利いてしまう（CTRL＋Cはちゃんと無効化される）というバグがあって，ちゃんとは使えない。

そういえば，Ver.2.03からはファイル名中に“—”（ハイフン）が利用できるようになった。プログラムのオプションと区別するためか，ファイル名の先頭での使用は許されないという小さな制限はあるようだが，とにかく，これでMS-DOSの“—”を含むファイルが読み書きできる[13]。もっとも，フリーソフトウェアにはHuman68kのファイル名の仕様を拡張するプログラムもあったし，“—”を使えるようにするだけならVer.2.01～2.02ならHuman68kに1バイトのパッチをあてるだけですむ。Human68k本体の“どこか”にある，

20 22 27 2B 2E

というバイト列の2Eを2Dに書き換えればよい。メモリ上で見つけたら，Human68kの先頭番地とXファイル先頭のヘッダの大きさを考慮すれば，ファイル上の書き換え位置もわかる。

あとは非公開コールがひとつある。冒頭の表1にも示したように，Human68k Ver.2.0で新設されたDOSコールのうち，コール番号FF7C$_H$は『プログラマーズマニュアル』では公開されなかった。一般アプリケーションからは使ってくれるな，ということらしい。が，このコールを使うと面白いこと（悪いこととも いう）がいろいろとできたりするので，簡単に紹介するだけしておこうと思う。実際に使うかどうかは読者の良識に委ねる。

このコールは，

```
move.w #ファイルハンドル,－(sp)
DOS    $ff7c
addq.l  #2,sp
```

のようにして呼び出すと，ファイルハンドルに対応したHuman68k内部のワークエリア先頭アドレスをd0.lに返す。ほかのDOSでの呼び方にしたがって，このワークエリアのことを以下FCB（File Control Block）と呼ぶことにする。

FCBはファイルの読み書きに必要な情報をひとまとめにしたものだ。FCB1個あたり96バイトを占め，その構造は表10のようになっている。解析が不十分なもので，途中の1バイトと末尾部分の用途は（未使用かどうかも含めて）特定できていない。意味がわかっている部分についても，あえて詳しく説明するのはやめておこう。興味のある人は，いろいろな状態のファイルのFCBを覗いてみてもらいたい。

＊　＊　＊

ということろで，何か足りないような気がしないでもないが，次回に続く。次回はHuman68k Ver.2.0の残りの部分，バックグラウンドプロセスを中心とする，プロセス/メモリ関係を片づける。

表10　FCBの構造（参考）

| | | |
|---|---|---|
| 00$_H$ | 1b | このFCBに対応しているファイルハンドルの数 |
| 01$_H$ | 1b | 装置情報 |
| 02$_H$ | 1l | 内部DPBへのポインタ/デバイスドライバへのポインタ |
| 06$_H$ | 1l | ファイルポインタ |
| 0A$_H$ | 1l | 排他制御情報へのポインタ |
| 0E$_H$ | 1b | アクセスモード |
| 0F$_H$ | 1b | ディレクトリ上の位置（セクタ先頭からの個数） |
| 10$_H$ | 1b | アクセス中のクラスタ中のセクタ |
| 11$_H$ | 1b | ??? |
| 12$_H$ | 1w | アクセス中のクラスタ番号 |
| 14$_H$ | 1l | アクセス中のセクタ番号 |
| 18$_H$ | 1l | I/Oバッファ先頭 |
| 1C$_H$ | 1l | 対応するディレクトリセクタ番号 |
| 20$_H$ | 1l | ファイルポインタ移動の上限 |
| 24$_H$ | 8b | ファイル名 |
| 2C$_H$ | 3b | ファイル名拡張子 |
| 2F$_H$ | 1b | ファイル属性 |
| 30$_H$ | 10b | ファイル名残り |
| 3A$_H$ | 1w | 先頭のFAT番号 |
| 3C$_H$ | 1w | 更新最終年月日 |
| 3E$_H$ | 1w | 更新最終時刻 |
| 40$_H$ | 1l | ファイルサイズ |
| 44$_H$ | 6l | ??? |

# THE SENTINEL

〈対応機種一覧〉●MZ-80K/C/700/1500●MZ-80B/2000●MZ-2500/2861●X1●X1turbo/Z●PC-8001/8801/88●SMC-777/C●PASOPIA/5●PASOPIA 7●FM-7/77/AV●PC-286/386/486/9801/98/9821●X68000
掲載されたプログラムの利用には各機種用のS-OS"SWORD"システムが必要です。

## 第128部　EDC-Tの拡張

**●EDC-T**

今月は11月号で呼びかけたUPDATE.$$$の出力ルーチンを，EDC-Tに拡張するプログラムです。UPDATE.$$$は，MAKEを使ううえで必要な，タイムスタンプの代わりとして使われるテンポラリファイルです。

呼びかけに応じてくれたのは，Small-C用SLANGコンパチ関数でお馴染みの伊藤直也氏。しかも，投稿原稿にあった日付は10月27日！　11月号の発売日が10月18日ですから，実に9日間でこの拡張プログラムを作ってしまったということになります。実質的には，もう少し短いかもしれませんので，この行動力には驚かされますね。

また，今月のプログラムによってUPDATE.$$$の出力ルーチンが付加されただけでなく，いくつかの機能強化がなされています。

強化されたのはEDC-Tでサポートされたタブコード変換部分です。EDC-Tでは，入力されたテキストにあるスペースを，すべてタブコードに変換してしまいました。確かに，タブコードに変換することでテキストのサイズを小さくすることができるのは，非常にありがたいものです。

しかし，問題となったのはアセンブラなどで文字列を定義しようとした場合，その中にあるスペースまでをタブコードに変換してしまうところです。S-OSでタブコードを正式にサポートしていればあまり問題がないのですが，あいにくタブコードはアプリケーション側で判別する必要があります。いままで，EDC-Tで文字列中のスペースを記述するためには，スペースをコード$20_H$に置き換えなくてはならなかったのです。

文字列中のスペースをタブコードに変換しない機能のほかにも，タブコードを使わないようにしてテキストをエディットしたり，タブコード←→スペースの相互変換機能，テキストのセーブ，ロード時に最近エディットしたファイル名を表示することもできるようになっており，なかなか使い勝手も向上しています。

かゆいところに手が届くEDC-T拡張プログラム。EDC-Tを使用している人は，ぜひ打ち込んでみてください。

**●改造のススメ**

今月号ではアプリケーションの拡張がなされました。S-OSの世界では，アプリケーションのみならず，S-OS自身も改良を重ねてきています。トランジェントコマンドやRAMディスクなどの拡張がそうです。最近では，オペレーション環境をまるごと置き換えた，DOSライクな操作体系をもつCOMMAND.OBJも発表されています。

これらの改造の理由は，使っているうちに不満が出てきた，もっと自分好みのものを作りたいという理由からでしょう。つまり，足りない部分は自分で補うという能動的な精神によって行われてきたのです。

S-OSの世界では基本ともいえるこの精神を，ぜひ皆さんも忘れないでいただきたいですね。




**●S-OSの系譜（40）**

1990年2月号では，1989年6月号で発表されたTTC（Tiny Tiny Compiler）をバージョンアップしたTTC++（TTCインクリメント）が発表されました。作者はTTCと同じ平井真二氏。TTIを含めたTT?シリーズでお馴染みですね。

このTT?シリーズの特徴はなんといっても名前のとおりのコンパクトさにあります。エディタ，インタプリタ(TTI)，コンパイラにランタイムルーチンを合わせて7Kバイトというものでしたから，TTIでプログラムを制作し，速度的に不満があればTTC++でコンパイルするようなことが簡単に行えるのです。

機能的に見ると，SLANGやSmall-Cなどの本格的なコンパイラに比べ，貧弱さは感じずにはいられません。それでも，アセンブラとのリンクも可能な命令をもっており，アセンブラでは記述が面倒臭いちょっとしたプログラム作成に威力を発揮するものでした。

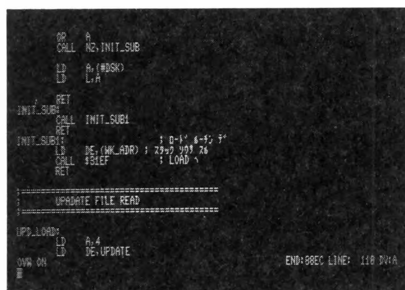
**1992■インデックス**

全機種共通
S-OS"SWORD"要

# EDC-Tの拡張

Itou Naoya
伊藤 直也

**今月は，11月号で提案したUPDATE.$$$出力ルーチンを追加するためのEDC-T拡張プログラムです。出力ルーチンだけでなく，タブコード関係の処理も改善がなされているので，ぜひ拡張してみてください。**



今回発表するプログラムは，1990年11月号で発表されたスクリーンエディタEDC-Tを拡張するものです。EDC-Tは，E-MATEの機能,操作体系を真似たスタンダードなスクリーンエディタでした。しかし，先月発表されたMAKEプログラムを使うために，対応しなくてはならない部分が出てきました。そこで，この拡張部分とEDC-Tにあるタブコード機能を強化してしまおう，というのがこのプログラムの目的です。

## 入力方法

まず,拡張プログラムのダンプリスト(リスト1)を，MACINTO-Cなどのツールを使って入力します。CRCチェックサムが合っているか確認したあと，$3000_H$～$5018_H$までをいったんデバイスにセーブしてください。そして，改造していないEDC-T，拡張プログラムの順にメモリへロードし，

#J5100

で実行して,EDC-Tに拡張プログラムが使えるようにパッチを当てます。

パッチを当てたら，

#S　EDC_T+．OBJ：3000：5018

として拡張部分と一緒にセーブします。ファイル名はなんでもいいのですが，COMMAND.OBJを使用する場合は，ファイル名にスペース，“－”(マイナス)，“/”記号が使えません（オプションと判断されてしまう）ので注意してください。

また，以上の作業はCOMMAND.OBJ上で行うことができません。必ずS-OSのモニタ上で行ってください。

## プログラムについて

冒頭で述べたとおり，この拡張プログラムは1992年11月号のTHE SENTINELで提案されている「UPDATE.$$$の出力ルーチン」が加えてあります。仕様もほぼ同じで，コマンドラインでのファイルオプション指定，メモリ内でのファイル名追加，エディタ終了時にUPDATE.$$$を出力する，となっています。

仕様が違うところも若干あります。それは更新の仕方です。11月号では，テキストのロード時に更新をするようになっていましたが，このプログラムではテキストのセーブ時に更新するようになっているのです。どちらにせよ出力結果は同じなので，問題はないでしょう。

そして，既存のL,S,Hコマンドなどの拡張や，小文字にも対応させせました。

## プログラムの起動方法

このプログラムによって，EDC-TにもUPDATE.$$$の出力機能,コマンドラインからのファイル指定つき起動ができるようになりました。基本的に起動方法は，EDITと同じようなものなのですが，もう一度説明しておきましょう。ここでは，COMMAND.OBJ上から起動することを前提としていますので，ファイル名の“－”キャラクタを“_”としています。また，COMMAND.XもしくはS-OSの拡張を行っていない人は，EDC_Tの部分を，J3000と置き換えてください。

1)　EDC_T

通常の起動方法です。エディットしたテキストファイル名は,UPDATE.$$$に更新されます。

2)　EDC_T　/N

EDC-Tでエディットしたテキストファイル名を終了時にUPDATE.$$$へ出力しないようにします。

3)　EDC_T　［ファイル名］

［ファイル名］のテキストを起動時に読み込みます。エディタ終了時にUPDATE.$$$は更新されます。

4)　EDC_T　［ファイル名］　/N

3)と同じように[ファイル名]のテキストを起動時に読み込みます。ただし，エディタ終了時にUPDATE.$$$を更新しません。

また,UPDATE.$$$のバッファについて説明しておきます。バッファはメモリ上に持っていて，起動時にこのバッファをオーバーするようなUPDATE.$$$を読み込んだ場合，バッファの内容はクリアされてしまいます。Sコマンド実行時にバッファをオーバーした場合は，セーブしようとしたファイル名が追加されないだけになります。この2点だけは注意してください。

## 拡張コマンド

今度は，EDC-T自身に拡張，追加されたコマンドについて説明します。

**・S（セーブ）コマンド**
**・L（ロード）コマンド**

ファイル名を省略した場合，最近指定したファイル名を表示するようになります。これは，EDITやREDAのエディタと同じような機能です。たとえば，現在エディットしているファイル名が“GAME.S”とすると，Sコマンドを入力したときに，

S　GAME.S

と表示されるようになるのです。

**・M（タブ制御）コマンド**

これは新しく追加されたコマンドです。コマンドを実行するたびにタブ制御モードが，ON→AUTO→OFF→ON……変更されます。タブ制御モードのそれぞれの機能は表1にまとめておきました。これは，EDC-Tのかぎりなく手抜きとしかいいようのない仕様の，タブコード変換機能をサポートするものです。表1を見てもらえばわかるとおり，AUTOモードを使うことで従来は記述不可能であった，文字列データ中にスペースを使用できるようになります。

**・H（タブ，スペース変換）コマンド**

Mコマンドで設定したタブ制御モードに従ってタブコード，スペースの変換を行います。それぞれのモードでどのような動作をするか，表2にまとめておきました。

## 拡張が終わって

本当は，もっと小さなプログラムにする予定でしたが，ずいぶんと大きなものになってしまいました。いままで不満のあった機能の拡張まで一緒にやってしまったから当然かもしれませんけどね。

今回は，既存のものに対して外部ファイル出力機能を追加するということでしたが，さらに新しい機能をもった，現在のS-OSの環境から一歩抜け出したエディタが登場したら，また楽しいでしょうね。ごくスタンダードなものでも，今回のようなちょっとした機能によって，ずいぶんと使い勝手がよくなるんですから。皆さんも，自分の使っているエディタの不満点を解消して，よりよいプログラミング環境を作りましょう。

**表1　タブ制御モード**

| モード名 | 機　　能 |
|---|---|
| ON | 編集時にテキストのスペースをタブコードに変換します |
| AUTO | 基本的にはモードONと同じですが，ダブルクォーテーション，シングルクォーテーション以降は，その行の終わりまでスペースをタブコードに変換しません |
| OFF | 編集時のスペースはそのままです。タブコードをサポートしていない言語のソースリストを記述するときに有効です |

**表2　Hコマンド**

| 制御モード | 機　　能 |
|---|---|
| ON | 制御モードOFFやタブコードをサポートしていないエディタで編集したテキストのスペースをタブコードに変換します |
| AUTO | 基本的には制御モードONと同じですが，ダブルクォーテーション，シングルクォーテーション以降は，その行の終わりまでスペースをタブコードに変換しません |
| OFF | テキスト中に存在するタブコードをスペースに変換します |

**リスト1**

```
4A81 CD F8 3F 21 19 4E 22 14 : C2
4A89 4E 22 16 4E 36 00 AF 32 : EB
4A91 9B 4D 3C 32 9A 4D 21 C5 : 23
4A99 4D 11 B0 4D 01 07 00 ED : 50
4AA1 B0 2A 76 1F 23 23 7E 23 : 56
4AA9 B7 28 36 FE 0D 28 32 FE : 78
4AB1 20 20 F3 CD 70 4D 22 98 : 77
4AB9 4D 7E 23 B7 28 23 FE 0D : FB
4AC1 28 1F FE 2D 28 0B FE 2F : D2
4AC9 28 07 3E 01 32 9B 4D 18 : A0
4AD1 E8 7E 2B 36 00 FE 4E 28 : 3B
4AD9 04 FE 6E 20 04 AF 32 9A : 0F
4AE1 4D 3A 9A 4D B7 C4 01 4B : 35
4AE9 3A 9B 4D B7 C4 F5 4A 3A : 16
4AF1 5D 1F 6F C9 CD F9 4A C9 : 8D
4AF9 ED 5B 98 4D CD EF 31 C9 : E3
---------------------------------
SUM: E4 59 C6 2D 25 51 53 DE 1977

4B01 3E 04 11 CC 4D CD A3 1F : FB
4B09 CD 09 20 D8 21 FE 01 ED : DB
4B11 5B 72 1F ED 52 38 1C 2A : A9
4B19 14 4E 19 2B 22 16 4E 2A : 56
4B21 14 4E 22 70 1F CD A6 1F : A5
4B29 38 04 CD 05 4C C9 11 D7 : 0B
4B31 4D 18 03 11 FF 4D 21 19 : FF
4B39 4E 22 14 4E 22 16 4E 36 : 8E
4B41 00 CD E5 1F CD C4 1F CD : 4E
4B49 BC 31 C9 ED 5B 14 4E 1A : 7A
4B51 13 B7 28 2C FE 0D 28 F7 : 48
4B59 21 B0 4D BE 23 20 16 1A : 4F
4B61 13 B7 28 0A FE 0D 28 06 : 35
4B69 BE 23 28 F3 18 07 7E B7 : 50
4B71 C8 FE 0D C8 1B 1A 13 B7 : 9A
4B79 28 06 FE 0D 28 D1 18 F5 : 3F
---------------------------------
SUM: 12 9C ED 58 10 16 B0 06 C561

4B81 CD AD 4B 38 15 11 B0 4D : 20
4B89 2A 16 4E 1A 13 77 23 B7 : 0C
4B91 20 F9 2B 22 16 4E CD 05 : 9C
4B99 4C C9 3E 0C CD F4 1F 11 : 50
4BA1 FF 4D CD E5 1F CD C4 1F : CD
4BA9 CD BC 31 C9 11 B0 4D 2A : BB
4BB1 16 4E 1A 13 23 B7 20 FA : 85
4BB9 ED 5B 14 4E 2B ED 52 11 : 25
4BC1 FD 01 ED 52 3F C9 3A 9A : 19
4BC9 4D B7 C8 3E 04 11 CC 4D : 38
4BD1 CD A3 1F 2A 16 4E ED 5B : 65
4BD9 14 4E ED 52 23 22 72 1F : 77
4BE1 21 00 00 22 70 1F 22 6E : 62
4BE9 1F CD AF 1F 38 0C 2A 14 : 3C
4BF1 4E 22 70 1F CD AC 1F 38 : CF
4BF9 01 C9 11 F0 4D CD E5 1F : E9
---------------------------------
SUM: EC 98 1F EB C7 D9 F7 A8 B8F1

4C01 CD C4 1F C9 2A 16 4E 2B : 32
4C09 7E FE 0D C8 23 36 0D 23 : DA
4C11 36 00 22 16 4E C9 CD E7 : 39
4C19 3F 1A FE 41 38 07 FE 61 : 36
4C21 38 03 D6 20 37 3F C9 CD : 3D
4C29 D3 1F CD 17 4C C9 FE 61 : 4A
4C31 38 03 D6 20 37 3F CD CA : 3E
4C39 31 C9 3A 9E 4D 3C FE 03 : 5C
4C41 38 01 AF 32 9E 4D CD 91 : 63
4C49 4C FE 01 28 10 FE 02 28 : AB
4C51 18 21 62 3F 36 7E 23 36 : E7
4C59 FE 23 36 0D C9 3E C3 32 : 60
4C61 62 3F 21 76 4C 22 63 3F : 48
4C69 C9 3E C3 32 62 3F 21 86 : 44
4C71 4C 22 63 3F C9 7E FE 0D : 62
4C79 CA 8C 3F FE 22 CA 86 4C : 51
---------------------------------
SUM: 0F 38 CD 68 20 4F 75 D0 73B8

4C81 FE 27 C2 67 3F 7E FE 0D : 16
4C89 CA 8C 3F 12 13 23 18 F5 : EA
4C91 3A 9E 4D FE 03 38 04 AF : 11
4C99 32 9E 4D 11 9F 4D B7 28 : F9
4CA1 0A 11 A4 4D FE 01 28 03 : 36
4CA9 11 A9 4D 21 04 16 CD 1E : 2D
4CB1 20 CD E5 1F C9 3A 5D 1F : 70
4CB9 32 9C 4D 1B 1A 32 AE 4D : 7D
4CC1 13 AF 32 9D 4D 21 B0 4D : FC
4CC9 06 14 CD E7 3F 1A 13 4E : 88
4CD1 77 23 B7 28 21 FE 0D 28 : CD
4CD9 1D FE 2E 20 07 78 FE 05 : EB
4CE1 38 02 06 05 FE 3A 20 0B : A8
4CE9 3A 9D 4D B7 20 08 3E 01 : 42
4CF1 32 9D 4D 05 20 D4 2B 36 : 76
4CF9 00 11 B0 4D 1A B7 28 06 : 0D
---------------------------------
SUM: F2 43 F2 0A E5 27 50 76 DBB5

4D01 3E 0C CD F4 1F C9 79 12 : 7E
4D09 B7 20 0B 21 C5 4D 11 B0 : D6
4D11 4D 01 07 00 ED B0 21 00 : 13
4D19 17 CD 1E 20 11 AE 4D CD : FB
4D21 E5 1F 21 AE 4D CD A1 3E : CC
4D29 AF 32 3E 42 21 02 17 22 : BD
4D31 3F 42 CD DC 35 11 97 44 : 4B
4D39 CD 17 4C 1A 13 FE 4C CA : 71
4D41 B6 4C FE 53 CA B6 4C E1 : 00
4D49 E1 C9 CD A6 1F F5 3A 9C : 07
4D51 4D 32 5D 1F F1 C9 CD AC : 2E
4D59 1F F5 D4 4C 4B 3A 9C 4D : A2
4D61 32 5D 1F F1 C9 CD BC 31 : 22
4D69 3A 9C 4D 32 5D 1F C9 7E : 18
4D71 23 FE 20 28 FA 2B C9 D6 : 2D
4D79 30 BE 30 12 31 EE 30 02 : 81
---------------------------------
SUM: BB 95 2D DC 0E 05 00 FA D70E

4D81 31 49 32 E6 31 7B 31 DA : 49
4D89 31 2F 34 73 33 94 32 4A : 4A
4D91 31 5A 31 46 33 3B 4C 00 : BC
4D99 00 00 00 00 00 00 4F 4E : 9D
4DA1 20 20 00 41 55 54 4F 00 : 79
4DA9 4F 46 46 20 00 20 20 00 : 3B
4DB1 00 00 00 00 00 00 00 00 : 00
4DB9 00 00 00 00 00 00 00 00 : 00
4DC1 00 00 00 00 4E 4F 4E 41 : 2C
4DC9 4D 45 00 55 50 44 41 54 : 10
4DD1 45 2E 24 24 24 00 55 50 : 84
4DD9 44 41 54 45 20 46 69 6C : 59
4DE1 65 20 52 65 61 64 20 45 : 66
4DE9 72 72 6F 72 21 0D 00 43 : 36
4DF1 61 6E 27 74 20 55 50 44 : 73
4DF9 41 54 45 21 0D 00 55 50 : AD
---------------------------------
SUM: 51 40 82 2A 7D 5D 7F DF AD04

4E01 44 41 54 45 20 42 75 66 : 5B
4E09 66 65 72 20 4F 76 65 72 : F9
4E11 21 0D 00 19 4E 19 4E 0D : 09
---------------------------------
SUM: CB B3 C6 7E BD D1 28 E5 DE7F
```

$4E19_H$～$50FF_H$まで0で埋める

```
5100 21 81 4A 22 58 30 3E CD : A1
5108 32 5A 30 21 62 30 22 5B : EC
5110 30 AF 32 5D 30 21 C7 4B : D1
5118 22 5F 30 21 28 4C 22 AA : 12
5120 31 AF 32 AC 31 21 B6 4C : 12
5128 22 F2 31 22 55 32 21 4B : 5A
5130 4D 22 31 32 21 57 4D 22 : B9
5138 8A 32 21 66 4D 22 3B 32 : 1F
5140 22 91 32 21 17 4C 22 84 : 0F
5148 30 AF 32 86 30 21 2F 4C : 63
5150 22 DC 31 3E E1 32 83 32 : 35
5158 21 38 07 22 84 32 21 91 : EA
5160 4C 22 1D 40 AF 32 82 40 : 6E
5168 32 67 42 32 70 42 3E 57 : 54
5170 32 66 42 21 4D FF 22 93 : FC
5178 35 21 78 4D 22 B2 30 21 : 40
---------------------------------
SUM: 49 42 46 0E 40 8F AF E6 1F6B

5180 00 51 22 06 30 22 41 42 : 4E
5188 C9                      : C9
---------------------------------
SUM: C9 51 22 06 30 22 41 42 E197
```

## リスト2

```
0000                 ;=========================================
0000                 ;          EDC-T カクチョウ プログラム
0000                 ;=========================================
0000
0000                         OFFSET          $9000-$4A81
4A81                         ORG     $4A81
4A81
1F5D P               #DSK    EQU     $1F5D
1F6E P               #EXADR  EQU     $1F6E
1F70 P               #DTADR  EQU     $1F70
1F72 P               #SIZE   EQU     $1F72
1F76 P               #KBFAD  EQU     $1F76
1FA3 P               #FILE   EQU     $1FA3
1FA6 P               #RDD    EQU     $1FA6
1FAC P               #WRD    EQU     $1FAC
1FAF P               #WOPEN  EQU     $1FAF
1FC4 P               #BELL   EQU     $1FC4
1FD3 P               #GETL   EQU     $1FD3
1FE5 P               #MSX    EQU     $1FE5
1FF4 P               #PRINT  EQU     $1FF4
2009 P               #ROPEN  EQU     $2009
201E P               #LOC    EQU     $201E
3F62 P               TBC2    EQU     $3F62
3F8C P               TBCEND  EQU     $3F8C
3FE7 P               SPCUT   EQU     $3FE7
4A81
01FE P               BUFMAX  EQU     510
4A81
4A81                 ;=========================================
4A81                 ;          タチアゲ チョクゴ ノ ショリ
4A81                 ;=========================================
4A81
4A81                 INIT:
4A81 CD F8 3F                CALL    $3FF8         ;#TENDCHK
4A84
4A84 21 19 4E                LD      HL,UPDBUF
4A87 22 14 4E                LD      (BUFTOP),HL
4A8A 22 16 4E                LD      (BUFEND),HL
4A8D 36 00                   LD      (HL),0            ; Buf CLR
4A8F
4A8F AF                      XOR     A
4A90 32 9B 4D                LD      (LOADFLG),A   ; FLG = 0
4A93 3C                      INC     A
4A94 32 9A 4D                LD      (UPDFLG),A    ; FLG = 1
4A97
4A97 21 C5 4D                LD      HL,FILEINIT
4A9A 11 B0 4D                LD      DE,FILENAME
4A9D 01 07 00                LD      BC,7
4AA0 ED B0                   LDIR                  ;'NONAME' COPY
4AA2
4AA2
4AA2 2A 76 1F                LD      HL,(#KBFAD)
4AA5 23                      INC     HL
4AA6 23                      INC     HL            ; Skip Prompt
4AA7                 INIT_01:
4AA7 7E                      LD      A,(HL)
4AA8 23                      INC     HL
4AA9 B7                      OR      A
4AAA 28 36                   JR      Z,INIT_05 ; パラメータ ナシ
4AAC FE 0D                   CP      $0D
4AAE 28 32                   JR      Z,INIT_05 ; パラメータ ナシ
4AB0 FE 20                   CP      ' '
4AB2 20 F3                   JR      NZ,INIT_01
4AB4 CD 70 4D                CALL    SPCUT2
4AB7 22 98 4D                LD      (WK_ADR),HL
4ABA                 INIT_02:
4ABA 7E                      LD      A,(HL)
4ABB 23                      INC     HL
4ABC B7                      OR      A
4ABD 28 23                   JR      Z,INIT_05
4ABF FE 0D                   CP      $0D
4AC1 28 1F                   JR      Z,INIT_05
4AC3 FE 2D                   CP      '-'
4AC5 28 0B                   JR      Z,INIT_03
4AC7 FE 2F                   CP      '/'
4AC9 28 07                   JR      Z,INIT_03
4ACB
4ACB 3E 01                   LD      A,1
4ACD 32 9B 4D                LD      (LOADFLG),A
4AD0 18 E8                   JR      INIT_02
4AD2
4AD2                 INIT_03:
4AD2 7E                      LD      A,(HL)
4AD3 2B                      DEC     HL
4AD4 36 00                   LD      (HL),0
4AD6
4AD6 FE 4E                   CP      'N'
4AD8 28 04                   JR      Z,INIT_04
4ADA FE 6E                   CP      'n'
4ADC 20 04                   JR      NZ,INIT_05
4ADE                 INIT_04:
4ADE AF                      XOR     A
4ADF 32 9A 4D                LD      (UPDFLG),A
4AE2                 INIT_05:
4AE2 3A 9A 4D                LD      A,(UPDFLG)
4AE5 B7                      OR      A
4AE6 C4 01 4B                CALL    NZ,UPD_LOAD
4AE9 3A 9B 4D                LD      A,(LOADFLG)
4AEC B7                      OR      A
4AED C4 F5 4A                CALL    NZ,INIT_SUB
4AF0
4AF0 3A 5D 1F                LD      A,(#DSK)
4AF3 6F                      LD      L,A
4AF4
4AF4 C9                      RET
4AF5                 INIT_SUB:
4AF5 CD F9 4A                CALL    INIT_SUB1
4AF8 C9                      RET
4AF9                 INIT_SUB1:                    ; ロード ルーチン デ
4AF9 ED 5B 98 4D             LD      DE,(WK_ADR) ; スタック ソウサ スル
4AFD CD EF 31                CALL    $31EF         ; LOAD へ
4B00 C9                      RET
4B01
4B01                 ;=========================================
4B01                 ;          UPADATE FILE READ
4B01                 ;=========================================
4B01
4B01                 UPD_LOAD:
4B01 3E 04                   LD      A,4
4B03 11 CC 4D                LD      DE,UPDATE
4B06 CD A3 1F                CALL    #FILE
4B09 CD 09 20                CALL    #ROPEN
4B0C D8                      RET     C             ; ナイ
4B0D
4B0D 21 FE 01                LD      HL,BUFMAX
4B10 ED 5B 72 1F             LD      DE,(#SIZE)
4B14 ED 52                   SBC     HL,DE
4B16 38 1C                   JR      C,UPD_RERR2  ; バッファ オーバー
4B18
4B18 2A 14 4E                LD      HL,(BUFTOP)
4B1B 19                      ADD     HL,DE
4B1C 2B                      DEC     HL
4B1D 22 16 4E                LD      (BUFEND),HL
4B20
4B20 2A 14 4E                LD      HL,(BUFTOP)
4B23 22 70 1F                LD      (#DTADR),HL
4B26 CD A6 1F                CALL    #RDD
4B29 38 04                   JR      C,UPD_RERR1
4B2B CD 05 4C                CALL    ADDENDCD
4B2E C9                      RET
4B2F                 UPD_RERR1:
4B2F 11 D7 4D                LD      DE,ERRMSG1
4B32 18 03                   JR      UPD_RERR3
4B34                 UPD_RERR2:
4B34 11 FF 4D                LD      DE,ERRMSG3
4B37                 UPD_RERR3:
4B37 21 19 4E                LD      HL,UPDBUF
4B3A 22 14 4E                LD      (BUFTOP),HL
4B3D 22 16 4E                LD      (BUFEND),HL
4B40 36 00                   LD      (HL),0
4B42 CD E5 1F                CALL    #MSX
4B45 CD C4 1F                CALL    #BELL
4B48 CD BC 31                CALL    $31BC            ; SPWAIT
4B4B C9                      RET
4B4C
4B4C                 ;=========================================
4B4C                 ;          UPDATE FILE APPEND
4B4C                 ;=========================================
4B4C
4B4C                 APPEND:
4B4C ED 5B 14 4E             LD      DE,(BUFTOP)
4B50                 APPEND_1:
4B50 1A                      LD      A,(DE)
4B51 13                      INC     DE
4B52 B7                      OR      A
4B53 28 2C                   JR      Z,APPEND_5
4B55 FE 0D                   CP      $0D
4B57 28 F7                   JR      Z,APPEND_1
4B59
4B59 21 B0 4D                LD      HL,FILENAME
4B5C BE                      CP      (HL)          ; 1モジメ
4B5D 23                      INC     HL
4B5E 20 16                   JR      NZ,APPEND_4
4B60                 APPEND_2:
4B60 1A                      LD      A,(DE)
4B61 13                      INC     DE
4B62 B7                      OR      A
4B63 28 0A                   JR      Z,APPEND_3
4B65 FE 0D                   CP      $0D
4B67 28 06                   JR      Z,APPEND_3
4B69 BE                      CP      (HL)          ; 2モジメ イコウ
4B6A 23                      INC     HL
4B6B 28 F3                   JR      Z,APPEND_2
4B6D 18 07                   JR      APPEND_4
4B6F                 APPEND_3:
4B6F 7E                      LD      A,(HL)        ; サイゴ ノ モジ
4B70 B7                      OR      A
4B71 C8                      RET     Z
4B72 FE 0D                   CP      $0D
4B74 C8                      RET     Z
4B75 1B                      DEC     DE
4B76                 APPEND_4:
4B76 1A                      LD      A,(DE)
4B77 13                      INC     DE
4B78 B7                      OR      A
4B79 28 06                   JR      Z,APPEND_5
4B7B FE 0D                   CP      $0D
4B7D 28 D1                   JR      Z,APPEND_1
4B7F 18 F5                   JR      APPEND_4
4B81
4B81                 APPEND_5:
4B81 CD AD 4B                CALL    SIZE_CHECK
4B84 38 15                   JR      C,APPEND_ERR
4B86
4B86 11 B0 4D                LD      DE,FILENAME
4B89 2A 16 4E                LD      HL,(BUFEND)
4B8C                 APPEND_6:
4B8C 1A                      LD      A,(DE)
4B8D 13                      INC     DE
4B8E 77                      LD      (HL),A
4B8F 23                      INC     HL
4B90 B7                      OR      A
4B91 20 F9                   JR      NZ,APPEND_6
4B93
4B93 2B                      DEC     HL
4B94 22 16 4E                LD      (BUFEND),HL
4B97 CD 05 4C                CALL    ADDENDCD
4B9A C9                      RET
4B9B
4B9B                 APPEND_ERR:
4B9B 3E 0C                   LD      A,$0C
4B9D CD F4 1F                CALL    #PRINT
4BA0 11 FF 4D                LD      DE,ERRMSG3
4BA3 CD E5 1F                CALL    #MSX
4BA6 CD C4 1F                CALL    #BELL
4BA9 CD BC 31                CALL    $31BC            ; SPWAIT
4BAC C9                      RET
4BAD
4BAD                 SIZE_CHECK:
4BAD 11 B0 4D                LD      DE,FILENAME
4BB0 2A 16 4E                LD      HL,(BUFEND)
4BB3                 SIZE_CHECK1:
4BB3 1A                      LD      A,(DE)
4BB4 13                      INC     DE
4BB5 23                      INC     HL
4BB6 B7                      OR      A
4BB7 20 FA                   JR      NZ,SIZE_CHECK1
4BB9 ED 5B 14 4E             LD      DE,(BUFTOP)
4BBD 2B                      DEC     HL
4BBE ED 52                   SBC     HL,DE
4BC0 11 FD 01                LD      DE,BUFMAX - 1
4BC3 ED 52                   SBC     HL,DE
```



▶しばらく離れ気味になっていたパソコンでしたが，最近またやる気がフツフツとわいてきました。よし，やるべ！
国政　寛(21)京都府

```
4BC5 3F                  CCF
4BC6 C9                  RET
4BC7
4BC7           ;========================================
4BC7           ;         UPDATE FILE WRITE
4BC7           ;========================================
4BC7
4BC7           UPD_WRITE:
4BC7 3A 9A 4D            LD     A,(UPDFLG)
4BCA B7                  OR     A
4BCB C8                  RET    Z
4BCC
4BCC           UPD_WRITE1:
4BCC 3E 04               LD     A,4
4BCE 11 CC 4D            LD     DE,UPDATE
4BD1 CD A3 1F            CALL   #FILE
4BD4
4BD4 2A 16 4E            LD     HL,(BUFEND)
4BD7 ED 5B 14 4E         LD     DE,(BUFTOP)
4BDB ED 52               SBC    HL,DE
4BDD 23                  INC    HL
4BDE 22 72 1F            LD     (#SIZE),HL
4BE1 21 00 00            LD     HL,0
4BE4 22 70 1F            LD     (#DTADR),HL
4BE7 22 6E 1F            LD     (#EXADR),HL
4BEA
4BEA CD AF 1F            CALL   #WOPEN
4BED 38 0C               JR     C,UPD_WERR
4BEF
4BEF 2A 14 4E            LD     HL,(BUFTOP)
4BF2 22 70 1F            LD     (#DTADR),HL
4BF5 CD AC 1F            CALL   #WRD
4BF8 38 01               JR     C,UPD_WERR
4BFA C9                  RET
4BFB
4BFB           UPD_WERR:
4BFB 11 F0 4D            LD     DE,ERRMSG2
4BFE CD E5 1F            CALL   #MSX
4C01 CD C4 1F            CALL   #BELL
4C04 C9                  RET
4C05
4C05           ;========================================
4C05           ;         ADD'¥N' & '¥0'
4C05           ;========================================
4C05
4C05           ADDENDCD:
4C05 2A 16 4E            LD     HL,(BUFEND)
4C08 2B                  DEC    HL
4C09 7E                  LD     A,(HL)
4C0A FE 0D               CP     $0D
4C0C C8                  RET    Z
4C0D 23                  INC    HL
4C0E 36 0D               LD     (HL),$0D
4C10 23                  INC    HL
4C11 36 00               LD     (HL),0
4C13 22 16 4E            LD     (BUFEND),HL
4C16 C9                  RET
4C17
4C17           ;========================================
4C17           ;         ｺﾏﾝﾄﾞ  ｺﾓｼﾞ ﾁｪｯｸ
4C17           ;========================================
4C17
4C17           FROM_COM:
4C17 CD E7 3F            CALL   SPCUT
4C1A 1A                  LD     A,(DE)
4C1B FE 41               CP     'A'
4C1D 38 07               JR     C,COM_SKIP
4C1F FE 61               CP     'a'
4C21 38 03               JR     C,COM_SKIP
4C23 D6 20               SUB    $20
4C25 37                  SCF
4C26           COM_SKIP
4C26 3F                  CCF
4C27 C9                  RET
4C28
4C28           ;========================================
4C28           ;         DIR ｺﾏﾝﾄﾞ / LOAD / SAVE ｺﾓｼﾞ
4C28           ;========================================
4C28
4C28           FROM_DLS:
4C28 CD D3 1F            CALL   #GETL
4C2B CD 17 4C            CALL   FROM_COM
4C2E C9                  RET
4C2F
4C2F           ;========================================
4C2F           ;         ﾃﾞﾊﾞｲｽ ｺﾓｼﾞ ﾁｪｯｸ
4C2F           ;========================================
4C2F
4C2F           FROM_DEVICE:
4C2F FE 61               CP     'a'
4C31 38 03               JR     C,DV_SKIP
4C33 D6 20               SUB    $20
4C35 37                  SCF
4C36           DV_SKIP
4C36 3F                  CCF
4C37 CD CA 31            CALL   $31CA               ;  DVCHK
4C3A C9                  RET
4C3B
4C3B           ;========================================
4C3B           ;         Tab Code Mode Command
4C3B           ;========================================
4C3B
4C3B           TCM:
4C3B 3A 9E 4D            LD     A,(TCMSW)
4C3E 3C                  INC    A
4C3F FE 03               CP     3
4C41 38 01               JR     C,TCM_CNG
4C43 AF                  XOR    A
4C44           TCM_CNG:
4C44 32 9E 4D            LD     (TCMSW),A
4C47 CD 91 4C            CALL   TCM_PRINT
4C4A
4C4A FE 01               CP     1
4C4C 28 10               JR     Z,TCM1_SET
4C4E FE 02               CP     2
4C50 28 18               JR     Z,TCM2_SET
4C52
4C52           TCM0_SET:
4C52 21 62 3F            LD     HL,TBC2
4C55 36 7E               LD     (HL),$7E
4C57 23                  INC    HL
4C58 36 FE               LD     (HL),$FE
4C5A 23                  INC    HL
4C5B 36 0D               LD     (HL),$0D
4C5D C9                  RET
4C5E
4C5E           TCM1_SET:
4C5E 3E C3               LD     A,$C3
4C60 32 62 3F            LD     (TBC2),A
4C63 21 76 4C            LD     HL,TCM1_MAIN
4C66 22 63 3F            LD     (TBC2 + 1),HL
4C69 C9                  RET
4C6A
4C6A           TCM2_SET:
4C6A 3E C3               LD     A,$C3
4C6C 32 62 3F            LD     (TBC2),A
4C6F 21 86 4C            LD     HL,TCM2_MAIN
4C72 22 63 3F            LD     (TBC2 + 1),HL
4C75 C9                  RET
4C76
4C76           ;========================================
4C76           ;         ﾀﾌﾞｺｰﾄﾞ  MODE 1
4C76           ;========================================
4C76
4C76           TCM1_MAIN:
4C76 7E                  LD     A,(HL)
4C77 FE 0D               CP     $0D
4C79 CA 8C 3F            JP     Z,TBCEND        ;  RET Z
4C7C FE 22               CP     $22
4C7E CA 86 4C            JP     Z,TCM2_MAIN     ;  (") FIND
4C81 FE 27               CP     $27
4C83 C2 67 3F            JP     NZ,$3F67        ;  RET NZ
4C86
4C86                                            ;  (') FIND
4C86
4C86           ;========================================
4C86           ;         ﾀﾌﾞｺｰﾄﾞ  MODE 2
4C86           ;========================================
4C86
4C86           TCM2_MAIN:
4C86 7E                  LD     A,(HL)
4C87 FE 0D               CP     $0D
4C89 CA 8C 3F            JP     Z,TBCEND        ;  RET
4C8C
4C8C 12                  LD     (DE),A
4C8D 13                  INC    DE
4C8E 23                  INC    HL
4C8F 18 F5               JR     TCM2_MAIN
4C91
4C91           ;========================================
4C91           ;         TCM PRINT
4C91           ;========================================
4C91
4C91           TCM_PRINT:
4C91 3A 9E 4D            LD     A,(TCMSW)
4C94 FE 03               CP     3
4C96 38 04               JR     C,TCMP_SKIP
4C98 AF                  XOR    A
4C99 32 9E 4D            LD     (TCMSW),A
4C9C           TCMP_SKIP:
4C9C 11 9F 4D            LD     DE,TCMON
4C9F B7                  OR     A                      ; T = 0
4CA0 28 0A               JR     Z,TCM_PRINT2
4CA2 11 A4 4D            LD     DE,TCMAUTO
4CA5 FE 01               CP     1                      ; T = 1
4CA7 28 03               JR     Z,TCM_PRINT2
4CA9 11 A9 4D            LD     DE,TCMOFF  ; T = 2
4CAC           TCM_PRINT2:
4CAC 21 04 16            LD     HL,$1604
4CAF CD 1E 20            CALL   #LOC
4CB2 CD E5 1F            CALL   #MSX
4CB5 C9                  RET
4CB6
4CB6           ;========================================
4CB6           ;         ﾌｧｲﾙﾈｰﾑ ﾃﾝｿｳ
4CB6           ;========================================
4CB6
4CB6           FROM_LS:
4CB6 3A 5D 1F            LD     A,(#DSK)
4CB9 32 9C 4D            LD     (DSKWRK),A  ;ﾃﾞﾌｫﾙﾄ Drive
4CBC
4CBC 1B                  DEC    DE
4CBD 1A                  LD     A,(DE)
4CBE 32 AE 4D            LD     (COMNAME),A ;ｺﾏﾝﾄﾞ ﾎｿﾞﾝ
4CC1 13                  INC    DE
4CC2           MOVE_0:
4CC2 AF                  XOR    A
4CC3 32 9D 4D            LD     (CFLG),A    ;':' FLG
4CC6 21 B0 4D            LD     HL,FILENAME
4CC9 06 14               LD     B,20            ; Count
4CCB           MOVE_1:
4CCB CD E7 3F            CALL   SPCUT
4CCE 1A                  LD     A,(DE)
4CCF 13                  INC    DE
4CD0 4E                  LD     C,(HL)          ; ﾓﾄﾉ ﾓｼﾞ
4CD1 77                  LD     (HL),A
4CD2 23                  INC    HL
4CD3 B7                  OR     A
4CD4 28 21               JR     Z,MOVE_4
4CD6 FE 0D               CP     $0D
4CD8 28 1D               JR     Z,MOVE_4
4CDA FE 2E               CP     '.'
4CDC 20 07               JR     NZ,MOVE_2
4CDE 78                  LD     A,B
4CDF FE 05               CP     5
4CE1 38 02               JR     C,MOVE_2
4CE3 06 05               LD     B,5
4CE5           MOVE_2:
4CE5 FE 3A               CP     ':'
4CE7 20 0B               JR     NZ,MOVE_3
4CE9 3A 9D 4D            LD     A,(CFLG)
4CEC B7                  OR     A
4CED 20 08               JR     NZ,MOVE_4
4CEF 3E 01               LD     A,1
4CF1 32 9D 4D            LD     (CFLG),A
4CF4           MOVE_3:
4CF4 05                  DEC    B
4CF5 20 D4               JR     NZ,MOVE_1
4CF7           MOVE_4:
4CF7 2B                  DEC    HL
4CF8 36 00               LD     (HL),0       ; Put End Code
4CFA
4CFA 11 B0 4D            LD     DE,FILENAME
4CFD 1A                  LD     A,(DE)
4CFE B7                  OR     A
4CFF 28 06               JR     Z,DSPNAME
4D01
4D01 3E 0C               LD     A,$0C
```

▶セーラームーンのカレンダーを持って駐車場まで歩くのは，とても恥ずかしかった。でも，どっちかっていうとセーラーファッションのときのほうが……あ，僕はそういう趣味ないですよ，本当だってば。
中野　克己(24)岐阜県

```
4D03 CD F4 1F                   CALL   #PRINT
4D06 C9                         RET
4D07
4D07                    ;========================================
4D07                    ;       FILE NAME ﾋｮｳｼﾞ
4D07                    ;========================================
4D07
4D07                    DSPNAME:
4D07 79                         LD     A,C
4D08 12                         LD     (DE),A     ; File Name ﾓﾄﾍ
4D09 B7                         OR     A
4D0A 20 0B                      JR     NZ,DSPNAME_1
4D0C
4D0C 21 C5 4D                   LD     HL,FILEINIT
4D0F 11 B0 4D                   LD     DE,FILENAME
4D12 01 07 00                   LD     BC,7
4D15 ED B0                      LDIR              ;'NONAME' COPY
4D17                    DSPNAME_1:
4D17 21 00 17                   LD     HL,$1700
4D1A CD 1E 20                   CALL   #LOC
4D1D 11 AE 4D                   LD     DE,COMNAME
4D20 CD E5 1F                   CALL   #MSX            ; ﾋｮｳｼﾞ
4D23
4D23 21 AE 4D                   LD     HL,COMNAME
4D26 CD A1 3E                   CALL   $3EA1           ; GETLIN
4D29
4D29 AF                         XOR    A
4D2A 32 3E 42                   LD     ($423E),A  ; TEXTLFT
4D2D 21 02 17                   LD     HL,$1702
4D30 22 3F 42                   LD     ($423F),HL ; CURXY
4D33 CD DC 35                   CALL   $35DC           ; EDM2
4D36
4D36 11 97 44                   LD     DE,$4497   ; TEXTLFT
4D39 CD 17 4C                   CALL   FROM_COM
4D3C 1A                         LD     A,(DE)
4D3D 13                         INC    DE
4D3E FE 4C                      CP     'L'
4D40 CA B6 4C                   JP     Z,FROM_LS
4D43 FE 53                      CP     'S'
4D45 CA B6 4C                   JP     Z,FROM_LS
4D48
4D48 E1                         POP    HL         ; RET Adr Clear
4D49 E1                         POP    HL         ; RET Ade Clear
4D4A C9                         RET
4D4B
4D4B                    ;========================================
4D4B                    ;       ﾃﾞﾌｫﾙﾄ ﾄﾞﾗｲﾌﾞ ﾓﾄﾆ ﾓﾄﾞｽ
4D4B                    ;========================================
4D4B
4D4B                    FROM_LOAD:
4D4B CD A6 1F                   CALL   #RDD
4D4E F5                         PUSH   AF
4D4F 3A 9C 4D                   LD     A,(DSKWRK)
4D52 32 5D 1F                   LD     (#DSK),A
4D55 F1                         POP    AF
4D56 C9                         RET
4D57                    FROM_SAVE:
4D57 CD AC 1F                   CALL   #WRD
4D5A F5                         PUSH   AF
4D5B D4 4C 4B                   CALL   NC,APPEND
4D5E 3A 9C 4D                   LD     A,(DSKWRK)
4D61 32 5D 1F                   LD     (#DSK),A
4D64 F1                         POP    AF
4D65 C9                         RET
4D66
4D66                    ;========================================
4D66                    ;       LOAD SAVE / ERR ﾙｰﾁﾝ ｶﾗ
4D66                    ;========================================
4D66
4D66                    FROM_LS_ERR:
4D66 CD BC 31                   CALL   $31BC              ; SPWAIT
4D69 3A 9C 4D                   LD     A,(DSKWRK)
4D6C 32 5D 1F                   LD     (#DSK),A
4D6F C9                         RET
4D70
4D70                    ;========================================
4D70                    ;       ORIGINAL SPACE CUT
4D70                    ;========================================
4D70
4D70                    SPCUT2:
4D70 7E                         LD     A,(HL)
4D71 23                         INC    HL
4D72 FE 20                      CP     ' '
4D74 28 FA                      JR     Z,SPCUT2
4D76 2B                         DEC    HL
4D77 C9                         RET
4D78
4D78                    ;========================================
4D78                    ;       WORK AREA
4D78                    ;========================================
4D78
4D78                    COMTBL2:
4D78 D6 30                      DW     $30D6           ;  #TEDIT
4D7A BE 30                      DW     $30BE           ;  #TCOM
4D7C 12 31                      DW     $3112           ;  #ENTADRSET
4D7E EE 30                      DW     $30EE           ;  #TEXTCLR
4D80 02 31                      DW     $3102           ;  #TEXTREC
4D82 49 32                      DW     $3249           ;  #SAVE
4D84 E6 31                      DW     $31E6           ;  #LOAD
4D86 7B 31                      DW     $317B           ;  #DIRCOM
4D88 DA 31                      DW     $31DA           ;  #DEVICE
4D8A 2F 34                      DW     $342F           ;  #SEARCH
4D8C 73 33                      DW     $3373           ;  #CHANGE
4D8E 94 32                      DW     $3294           ;  #PRTOUT
4D90 4A 31                      DW     $314A           ;  #SCRMODE
4D92 5A 31                      DW     $315A           ;  #TABEX
4D94 46 33                      DW     $3346           ;  #ZERO
4D96 3B 4C                      DW     TCM
4D98                    WK_ADR:
4D98 00 00                      DW     $0000
4D9A                    UPDFLG:
4D9A 00                         DB     0
4D9B                    LOADFLG:
4D9B 00                         DB     0
4D9C                    DSKWRK:
4D9C 00                         DB     0
4D9D                    CFLG:
4D9D 00                         DB     0
4D9E                    TCMSW:
4D9E 00                         DB     0
4D9F                    TCMON:
4D9F 4F 4E 20 20                DB     "ON  ",0
4DA3 00
4DA4                    TCMAUTO:
4DA4 41 55 54 4F                DB     "AUTO",0
4DA8 00
4DA9                    TCMOFF:
4DA9 4F 46 46 20                DB     "OFF ",0
4DAD 00
4DAE                    COMNAME:
4DAE 20 20                      DB     $20,$20
4DB0                    FILENAME:
4DB0                            DS     21
4DC5                    FILEINIT:
4DC5 4E 4F 4E 41                DB     "NONAME",0
4DC9 4D 45 00
4DCC                    UPDATE:
4DCC 55 50 44 41                DB     "UPDATE.$$$",0
4DD0 54 45 2E 24
4DD4 24 24 00
4DD7                    ERRMSG1:
4DD7 55 50 44 41                DB     "UPDATE File Read Error!",$0D,0
4DDB 54 45 20 46
4DDF 69 6C 65 20
4DE3 52 65 61 64
4DE7 20 45 72 72
4DEB 6F 72 21 0D
4DEF 00
4DF0                    ERRMSG2:
4DF0 43 61 6E 27                DB     "Can^'t UPDATE!",$0D,0
4DF4 74 20 55 50
4DF8 44 41 54 45
4DFC 21 0D 00
4DFF                    ERRMSG3:
4DFF 55 50 44 41                DB     "UPDATE Buffer Over!",$0D,0
4E03 54 45 20 42
4E07 75 66 66 65
4E0B 72 20 4F 76
4E0F 65 72 21 0D
4E13 00
4E14                    BUFTOP:
4E14 19 4E                      DW     UPDBUF
4E16                    BUFEND:
4E16 19 4E                      DW     UPDBUF
4E18                    DUMMY:
4E18 0D                         DB     $0D
4E19                    UPDBUF:
4E19                            DS     BUFMAX + 2
5019
5019                    ;========================================
5019                    ;       ﾊﾟｯﾁ ｱﾃ ｼﾞｯｺｳ ﾌﾟﾛｸﾞﾗﾑ
5019                    ;========================================
5019
5100                            ORG    $5100
5100
5100
5100 21 81 4A                   LD     HL,INIT
5103 22 58 30                   LD     ($3058),HL
5106 3E CD                      LD     A,$CD
5108 32 5A 30                   LD     ($305A),A
510B 21 62 30                   LD     HL,$3062
510E 22 5B 30                   LD     ($305B),HL
5111 AF                         XOR    A
5112 32 5D 30                   LD     ($305D),A
5115 21 C7 4B                   LD     HL,UPD_WRITE
5118 22 5F 30                   LD     ($305F),HL
511B
511B 21 28 4C                   LD     HL,FROM_DLS
511E 22 AA 31                   LD     ($31AA),HL
5121 AF                         XOR    A
5122 32 AC 31                   LD     ($31AC),A
5125
5125 21 B6 4C                   LD     HL,FROM_LS
5128 22 F2 31                   LD     ($31F2),HL
512B 22 55 32                   LD     ($3255),HL
512E
512E 21 4B 4D                   LD     HL,FROM_LOAD
5131 22 31 32                   LD     ($3231),HL
5134
5134 21 57 4D                   LD     HL,FROM_SAVE
5137 22 8A 32                   LD     ($328A),HL
513A
513A 21 66 4D                   LD     HL,FROM_LS_ERR
513D 22 3B 32                   LD     ($323B),HL
5140 22 91 32                   LD     ($3291),HL
5143
5143 21 17 4C                   LD     HL,FROM_COM
5146 22 84 30                   LD     ($3084),HL
5149 AF                         XOR    A
514A 32 86 30                   LD     ($3086),A
514D
514D 21 2F 4C                   LD     HL,FROM_DEVICE
5150 22 DC 31                   LD     ($31DC),HL
5153                    ;
5153                    ;       S ｺﾏﾝﾄﾞ ﾃﾞﾊﾞｯｸﾞ
5153                    ;
5153 3E E1                      LD     A,$E1
5155 32 83 32                   LD     ($3283),A
5158 21 38 07                   LD     HL,$0738
515B 22 84 32                   LD     ($3284),HL
515E                    ;
515E                    ;       Tab Code Mode ｶﾝｹｲ ｽﾙﾓﾉ
515E                    ;
515E 21 91 4C                   LD     HL,TCM_PRINT
5161 22 1D 40                   LD     ($401D),HL
5164
5164 AF                         XOR    A
5165 32 82 40                   LD     ($4082),A
5168 32 67 42                   LD     ($4267),A
516B 32 70 42                   LD     ($4270),A
516E 3E 57                      LD     A,57H
5170 32 66 42                   LD     ($4266),A
5173
5173 21 4D FF                   LD     HL,$FF4D
5176 22 93 35                   LD     ($3593),HL
5179 21 78 4D                   LD     HL,COMTBL2
517C 22 B2 30                   LD     ($30B2),HL
517F                    ;
517F                    ;       TEXT TOP ADDRESS
517F                    ;
517F 21 00 51                   LD     HL,$5100
5182 22 06 30                   LD     ($3006),HL
5185 22 41 42                   LD     ($4241),HL
5188
5188 C9                         RET
5189
5189
```

## ▶ 全機種共通システムインデックス ◀

＊以下のアプリケーションは，基本システムであるS-OS "MACE" またはS-OS "SWORD" がないと動作しませんのでご注意ください。

**1985**

■85年6月号
序論　共通化の試み
第1部　S-OS "MACE"
第2部　Lisp-85インタプリタ
第3部　チェックサムプログラム
■85年7月号
第4部　マシン語プログラム開発入門
第5部　エディタアセンブラZEDA
第6部　デバッグツールZAID
■85年8月号
第7部　ゲーム開発パッケージBEMS
第8部　ソースジェネレータZING
■85年9月号
インタラプト　S-OS番外地
第9部　マシン語入力ツールMACINTO-S
第10部　Lisp-85入門(1)
■85年10月号
第11部　仮想マシンCAP-X85
連載　Lisp-85入門(2)
■85年11月号
連載　Lisp-85入門(3)
■85年12月号
第12部　Prolog-85発表

**1986**

■86年1月号
第13部　リロケータブルのお話
第14部　FM音源サウンドエディタ
■86年2月号
第15部　S-OS "SWORD"
第16部　Prolog-85入門(1)
■86年3月号
第17部　magiFORTH発表
連載　Prolog-85入門(2)
■86年4月号
第18部　思考ゲームJEWEL
第19部　LIFE GAME
連載　基礎からのmagiFORTH
連載　Prolog-85入門(3)
■86年5月号
第20部　スクリーンエディタE-MATE
連載　実戦演習magiFORTH
■86年6月号
第21部　Z80TRACER
第22部　magiFORTH TRACER
第23部　ディスクダンプ＆エディタ
第24部　"SWORD" 2000 QD
連載　対話で学ぶmagiFORTH
特別付録　PC-8801版S-OS "SWORD"
■86年7月号
第25部　FM音源ミュージックシステム
付録　FM音源ボードの製作
連載　計算力アップのmagiFORTH
特別付録　SMC-777版S-OS "SWORD"
■86年8月号
第26部　対局五目並べ
第27部　MZ-2500版S-OS "SWORD"
■86年9月号
第28部　FuzzyBASIC発表
連載　明日に向かってmagiFORTH
■86年10月号
第29部　ちょっと便利な拡張プログラム
第30部　ディスクモニタDREAM
第31部　FuzzyBASIC料理法<1>
■86年11月号
第32部　パズルゲームHOTTAN
第33部　MAZE in MAZE
連載　FuzzyBASIC料理法<2>
■86年12月号
第34部　CASL & COMET
連載　FuzzyBASIC料理法<3>

**1987**

■87年1月号
第35部　マシン語入力ツールMACINTO-C
連載　FuzzyBASIC料理法<4>
■87年2月号
第36部　アドベンチャーゲームMARMALADE
第37部　テキアベ作成ツールCONTEX
■87年3月号
第38部　魔法使いはアニメがお好き
第39部　アニメーションツールMAGE
付録　"SWORD" 再掲載とMAGICの標準化
■87年4月号
第40部　INVADER GAME
第41部　TANGERINE
■87年5月号
第42部　S-OS "SWORD" 変身セット
第43部　MZ-700用 "SWORD" をQD対応に
■87年6月号
インタラプト　コンパイラ物語
第44部　FuzzyBASICコンパイラ
第45部　エディタアセンブラZEDA-3
■87年7月号
第46部　STORY MASTER
■87年8月号
第47部　パズルゲーム碁石拾い
第48部　漢字出力パッケージJACKWRITE
特別付録　FM-7/77版S-OS "SWORD"
■87年9月号
第49部　リロケータブル逆アセンブラInside-R
特別付録　PC-8001/8801版S-OS "SWORD"
■87年10月号
第50部　tiny CORE WARS
第51部　FuzzyBASICコンパイラの拡張
第52部　X1turbo版S-OS "SWORD"
■87年11月号
序論　神話のなかのマイクロコンピュータ
付録　S-OSの仲間たち
第53部　もうひとつのFuzzyBASIC入門
第54部　ファイルアロケータ＆ローダ
インタラプト　S-OSこちら集中治療室
第55部　BACK GAMMON
■87年12月号
第56部　タートルグラフィックパッケージTURTLE
第57部　X1turbo版 "SWORD" アフターケア
　　　　ラインプリントルーチン
特別付録　PASOPIA7版S-OS "SWORD"

**1988**

■88年1月号
第58部　FuzzyBASICコンパイラ・奥村版
付録　石上版コンパイラ拡張部の修正
■88年2月号
第59部　シューティングゲームELFES
■88年3月号
第60部　構造型コンパイラ言語SLANG
■88年4月号
第61部　デバッギングツールTRADE
第62部　シミュレーションウォーゲームWALRUS
■88年5月号
第63部　シューティングゲームELFES II
第64部　地底最大の作戦
■88年6月号
第65部　構造化言語SLANG入門(1)
第66部　Lisp-85用NAMPAシミュレーション
■88年7月号
第67部　マルチウィンドウドライバMW-1
連載　構造化言語SLANG入門(2)
■88年8月号
第68部　マルチウィンドウエディタWINER
■88年9月号
第69部　超小型エディタTED-750
第70部　アフターケアWINERの拡張
■88年10月号
第71部　SLANG用ファイル入出力ライブラリ
第72部　シューティングゲームMANKAI
■88年11月号
第73部　シューティングゲームELFES IV
■88年12月号
第74部　ソースジェネレータSOURCERY

**1989**

■89年1月号
第75部　パズルゲームLAST ONE
第76部　ブロックゲームFLICK
■89年2月号
第77部　高速エディタアセンブラREDA
特別付録　X1版S-OS "SWORD"<再掲載>
■89年3月号
第78部　Z80用浮動小数点演算パッケージSOR
　　　　OBAN
■89年4月号
第79部　SLANG用実数演算ライブラリ
■89年5月号
第80部　ソースジェネレータRING
■89年6月号
第81部　超小型コンパイラTTC
■89年7月号
第82部　TTC用パズルゲームTICBAN
■89年8月号
第83部　CP/M用ファイルコンバータ
■89年9月号
第84部　生物進化シミュレーションBUGS
■89年10月号
第85部　小型インタプリタ言語TTI
■89年11月号
第86部　TTI用パズルゲームPUSH BON!
■89年12月号
第87部　SLANG用リダイレクションライブラリDIO.LIB

**1990**

■90年1月号
第88部　SLANG用ゲームWORM KUN
特別付録　再掲載SLANGコンパイラ
■90年2月号
第89部　超小型コンパイラTTC++
■90年3月号
第90部　超多機能アセンブラOHM-Z80
■90年4月号
第91部　ファジィコンピュータシミュレーションI-MY
■90年5月号
第92部　インタプリタ言語STACK
■90年6月号
第93部　リロケータブルフォーマットの取り決め
第94部　STACK用ゲームSQUASH!
第95部　X68000対応S-OS "SWORD"
特別付録　PC-286対応S-OS "SWORD"
■90年7月号
第96部　リロケータブルアセンブラWZD
■90年8月号
第97部　リンカWLK
■90年9月号
第98部　BILLIARDS
■90年10月号
第99部　ライブラリアンWLB
■90年11月号
第100部　タブコード対応エディタEDC-T
■90年12月号
第101部　STACKコンパイラ

**1991**

■91年1月号
第102部　ブロックアクションゲームCOLUMNS
■91年2月号
第103部　ダイスゲームKISMET
■91年3月号
第104部　アクションゲームMUD BALLIN'
■91年4月号
第105部　SLANG用カードゲームDOBON
■91年5月号
第106部　実数型コンパイラ言語REAL
■91年6月号
第107部　Small-C処理系の移植
■91年7月号
第108部　REALソースリスト編
■91年8月号
第109部　Small-Cライブラリの移植
■91年9月号
第110部　SLANG用NEWファイル出力ライブラリ
■91年10月号
第111部　Small-C活用講座（初級編）
■91年11月号
第112部　Small-C活用講座（応用編）
第113部　MORTAL
■91年12月号
第114部　Small-C SLANGコンパチ関数

# バックナンバー案内

BACK ISSUES

ここには1992年1月号から1992年12月号までをご紹介しました。現在1991年1，5，8，9，11，12，1992年1，4～12月号の在庫がございます。バックナンバーおよび定期購読の申し込み方法については，168ページを参照してください。

1992

### 1月号
特集　SX-WINDOWの未来

連載　響子 in CGわ～るど/DōGA CGA/大人のためのX68000/ハード工作/Z80's Bar/ショートプロ/吾輩はX68000である/ANOTHER CG WORLD/Computer Music入門/カードゲーム

●MAGIC用ゲーム　3D MAZE

●CM-300/500&LA音源の活用法

LIVE in '92　DRAGON SABER/すき/THE ENTERTAINER

THE SOFTOUCH　出たな!! ツインビー/ブリッツクリーク/飛翔鮫 他

全機種共通システム　パズルゲームLINER

### 2月号(品切れ)
特集　2Dグラフィックの拡張

連載　響子 in CGわ～るど/大人のためのX68000/マシン語プログラミング/ハード工作/ショートプロ/ANOTHER CG WORLD/Z80's Bar/吾輩はX68000である/Computer Music入門/カードゲーム

●TREND ANALYSIS

●MIRAGE Model Stuff/Press Conductor PRO-68K

LIVE in '92　ストリートファイターII/Tide Over

THE SOFTOUCH　ジェノサイド2/アルシャーク/コード・ゼロ 他

全機種共通システム　シミュレーションゲームPOLANYI

### 3月号（品切れ）
特集　SCSIの活用

連載　響子 in CGわ～るど/DōGA CGA/大人のためのX68000/Z80's Bar/ショートプロ/吾輩はX68000である/マシン語プログラミング/ハード工作/ANOTHER CG WORLD/Computer Music入門/カードゲーム

●Z-MUSIC支援ツール　ZPDCON.X

●Z's-EX用拡張コマンド　MASK_reverse.X

LIVE in '92　ギャラクシーフォース/君が代

THE SOFTOUCH　グラディウスII/レミングス/大戦略III'90/伊忍道

全機種共通システム　カードゲームKLONDIKE

### 4月号
特集　成熟するゲームと日本の文化

連載　よい子のSX-WINDOW/Z80's Bar/響子 in CGわ～るど/ショートプロ/吾輩はX68000である/ハード工作/ANOTHER CG WORLD/Computer Music入門

●発表　1991年度GAME OF THE YEAR

●バーコードバトラー

LIVE in '92　あじさいのうた/ショパン練習曲作品25-2へ短調/IT'S MAGIC

THE SOFTOUCH　ファーストクィーンII/マスターオブモンスターズII 他

全機種共通システム　実践Small-C(1)オプティマイザO80

### 5月号
特集　明日のための環境づくり

第7回　言わせてくれなくちゃだワ

連載　響子 in CGわ～るど/大人のためのX68000/Z80's Bar/ハード工作/ショートプロ/マシン語プログラミング/Computer Music入門/吾輩はX68000である

●製品紹介　MIDI音源　03R/W/MIC68K

LIVE in '92　フレンズ/Danger Line

THE SOFTOUCH　エイリアンシンドローム/苦胃頭捕物帳 他

全機種共通システム　実践Small-C(2)COMMAND.OBJ

### 6月号
特別企画　Oh!MZ,Oh!X10年間の歩み

特別付録　創刊10周年記念PRO-68K(5"2HD)

連載　響子 in CGわ～るど/大人のためのX68000/マシン語プログラミング/ハード工作/ショートプロ/ANOTHER CG WORLD/Z80's Bar/吾輩はX68000である/Computer Music入門

●新製品紹介　Z'sSTAFF PRO-68K ver.3.0

LIVE in '92　Shake the Street/Ancient relics

THE SOFTOUCH　スピンディジーII/ロイヤルブラッド/ライフ&デス 他

全機種共通システム　実践Small-C講座(3)COMMAND.OBJ2

### 7月号
特集　超空間美術論

特別付録　DōGA CGAシステム&お試しディスク(5"2HD)

連載　よいこのSX-WINDOW/響子 in CGわ～るど/Z80's Bar/ANOTHER CG WORLD/大人のためのX68000/Computer Music入門/ハード工作/ショートプロ

●試用レポート　V70アクセラレータボード

LIVE in '92　Bye Bye My Love/MATERIAL GIRL/ヴェクザシオン

THE SOFTOUCH　将棋聖天&棋太平68K/シムアース/太閤立志伝

全機種共通システム　実践Small-C講座(4)関数リファレンス

### 8月号
特集　プログラミング再入門

連載　響子 in CGわ～るど/吾輩はX68000である/よいこのSX-WINDOW/マシン語プログラミング/ハード工作/ANOTHER CG WORLD/大人のためのX68000/Computer Music入門/ショートプロ

●新製品紹介　MATIER/TG100/SOUND SX-68K

LIVE in '92　氷穴/ガラガラヘビがやってくる/風の贈り物

THE SOFTOUCH　三國志III/シムアース/ウルティマVI/バトルテック

全機種共通システム　実践Small-C講座(5)ワイルドカード　グラフィックライブラリGRAPH.LIB

### 9月号
特集　数値演算の熱い逆襲

連載　DōGA CGアニメーション講座/大人のためのX68000/響子 in CGわ～るど/吾輩はX68000である/ショートプロ/マシン語プログラミング/ハード工作/ANOTHER CG WORLD

●新製品紹介　MATIER/MIREGE Model Stuff

LIVE in '92　恋をしようよ Yeah! Yeah!/ゆめいっぱい

THE SOFTOUCH　ファイナルファイト/ライジングサン/ヨーロッパ戦線/シューティング68K GAMES

全機種共通システム　O-EDIT & MODCNV

### 10月号
特集　DTMへの招待

連載　DōGA CGアニメーション講座/大人のためのX68000/響子 in CGわ～るど/吾輩はX68000である/ショートプロ/マシン語プログラミング/ハード工作/ANOTHER CG WORLD

●試用レポート　X68000用CD-ROMドライブ

LIVE in '92　美少女戦士セーラームーン/笑顔を探して 他

THE SOFTOUCH　ポピュラスII/リーディングカンパニー/ネクタリス/サークII

全機種共通システム　実践Small-C講座(6)SLENDER HUL

### 11月号
特集　ゲームマネージメント

連載　DōGA CGアニメーション講座/大人のためのX68000/響子 in CGわ～るど/ショートプロ/よいこのSX-WINDOW/ハード工作/ANOTHER CG WORLD/Computer Music入門

●新製品紹介　CHART PRO-68K

LIVE in '92　ストリートファイターII/スーパーマリオ 他

THE SOFTOUCH　キャッスルズ/シュートレンジ/ポピュラスII/サンダーレスキュー

全機種共通システム　実践Small-C講座(7)EDIT

### 12月号
Oh!X5周年特別企画　ショートプロ大集合

連載　DōGA CGアニメーション講座/マシン語プログラミング/響子 in CGわ～るど/ショートプロ/よいこのSX-WINDOW/大人のためのX68000/ハード工作/Computer Music入門

●エレクトロニクスショウ'92

LIVE in '92　LAST CHRISTMAS/闇の血族/ユーフォリー

THE SOFTOUCH　デスブレイド/ムーンクレスタ&テラクレスタ/ふしぎの海のナディア/ロードス島戦記II 他

全機種共通システム　実践Small-C講座(8)MAKE

# THE USER'S WORKS

## ●R-DRIVER/WM_DRV

**今回はちょっと毛色を変えてX1用の音楽ドライバを紹介する。Oh!Xにはいくつかのドライバが投稿されてきたがそれらのなかでも傑出したものである。それぞれすでに多くのユーザーによって愛用されている。**

●

今回紹介するのはX1用の音楽ドライバだ。どちらも，MIDIまでサポートされ，両者ともOh!Xで発表したMIDIボードとRS-232Cインタフェイスから MIDI信号を出力するMIDIアダプタに対応している。どちらもソースリストを完全に公開している点まで似通っている。音源ドライバとしての仕様も似ているが，コマンドレベルでの実現のしかたにはそれぞれの個性が見られるようだ。

**●R-DRIVER**

では，まずR-DRIVERから見てみよう。

R-DRIVERにはFM音源＋PSGをサポートしたR-DRIVER ver.2.7とFM音源とMIDIをサポートしたR-DRIVER[MIDI]ver.1.2の2種類がある。ちなみにR-DRIVER［MIDI］のほうではPSG関係の処理が割愛されているが，実用上の問題はほとんどないと考えていいだろう。

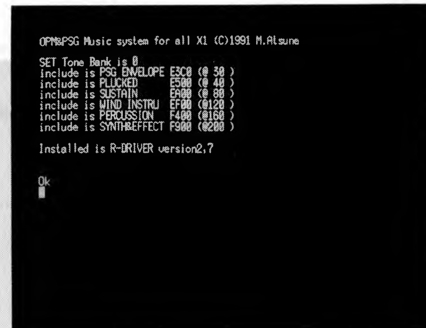
ドライバの性格としてはマイコンBASIC Magazine誌で発表されたNEW FM音源ドライバに近く，データの移行などはきわめて簡単に行うことができる。ただし，一般的なBASICのコマンドとして拡張されているのではなくUSR関数による記述となるので抵抗がある人もいるかもしれない。

なお,以下ではMIDI版の仕様をもとに解説していくことにする。

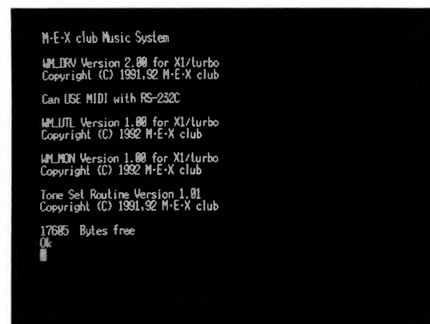
MIDI最大で16トラックのデータを同時シーケンス可能。ただし，処理速度を確保するため全音符の分解能は96に設定されている（非MIDI版は192)。



R-DRIVER［MIDI］



R-DRIVER ver.2.7



WM-DRV

最大の特徴は，シーケンスモードというものを持ち，バッファに登録したエンベロープパターンをボリュームをはじめ，パンポット，モジュレーション，アフタータッチ，そしてユーザー定義のコントロールチェンジに適用できるように設計されていることだ。Z-MUSIC ver.1.5での波形メモリ機能とほぼ同等のものが装備されているといっていい。これがあるおかげで，シンプルなコマンドもとてつもなく複雑なシーケンスをこなすようになっている。

そのほか，和音コマンドで1トラックでの和音出力（MIDIのみ）ができ，さらにディレイを指定してアルペジオ奏法もできる。NAGDRVと同様のユーザーマクロを装備し，MML記述を簡略化することも可能。

少し変わったところでは，GMレベル1に対応したコマンドを装備している，など。

また，R-DRIVER[MIDI]は近くver.2.0へのバージョンアップが予定されており，処理速度アップのほか，最大32トラックのシーケンス，波形メモリ的なエンベロープの指定に相対シーケンス記述方式などが加えられる予定となっている。

マニュアルには，RS-232Cを使ったMIDI出力用に超簡易インタフェイスの回路図も掲載されている。

連絡先は，

〒223　横浜市港北区日吉3-21-8-203
西尾将人方STONE-TONE

まで。62円切手と宛名カードを同封のこと。

**●WM_DRV**

WM_DRVはM·E·X clubで制作されたX1用の音楽ドライバである。こちらはMusicBASICとアッパーコンパチを保ちつつMIDI対応などを行ったBASIC拡張型の音楽ドライバとなっている。MusicBASICで作成された既存のデータが使用できる，操作法が同じなどの特徴を持つ。現在のバージョンは2.02となっている。

基本仕様はFM音源，PSG，MIDIをすべてサポートした16トラック同時シーケンスのドライバで，全音符の分解能は基本的には192である。コマンド体系を見ると，MusicBASICをもとにZ-MUSIC風のMML記述を取り入れたような仕様である。ベロシティシーケンスやダンパーコマンドまでが実現されているのには驚いた（FM音源でも使用可能)。ダンパーを使いFM音源でも和音発生をサポートしていたりもするほか，R-DRIVER同様，NAGDRV式の簡易マクロ機能も装備されている。

独自のものとしては，疑似リバーブコマンドや新型のPSGエンベロープなどがある。とんでもないのが，PSGサンプリングだ。A/Dコンバータからデジタル録音されたディスク3枚分のPCMデータが用意され，PSGを使って出力する，という仕様だ。このときMMLによる音量変更，ピッチ変更までサポートされている。かなり力業の感のある機能である。メモリや処理速度の関係から多用はできないが，ドラムなどで使う分には実用に耐えるものとなっている。

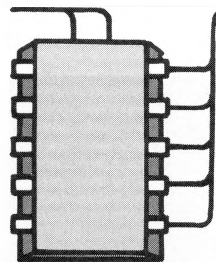
かなり処理が重くなるようだが，このドライバでは全音符の分解能を変えて処理を軽減することもできるようになっている（当然演奏は少し粗くなるが)。

サポートツールもかなり揃っているようだ。

連絡先は，

〒851-11　長崎県長崎市小江原町358-7
増山修方M·E·X club

まで。62円切手と宛名カードを同封のこと。



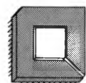
コンピュータアーキテクチャ編

# レジスタ加算器の製作

Misawa Kazuhiko
三沢 和彦

レジスタ加算器製作実習と個々の部品について詳しい解説を行います。TTL ICの選び方，回路を製作するうえでの注意点など，まだ製作に慣れていない人はゆっくり読み進めてください。

先月号ではアキュムレータ付き2桁加算器の設計を行いました。アキュムレータを付け加えることで，3つ以上の数値データを順次加えていくことができるようになります。回路図は先月示したとおりですが，今月はまず個々の部品について詳細にあたりながら，実際に製作実習を行っていく予定です。



## TTL ICの選び方

実際の回路図をもう一度図1に，必要な部品を表1に示します。まずは個々の部品についてそれぞれ見ていきますが，最も重要なICの説明から入りましょう。

TTL ICはLS183とALS74とを使用します。LS183は前回の加算器にも使ったALUパッケージで，繰り上がり付きの1桁加算器が2個，ひとつのパッケージに入っているものです。問題なのは，アキュムレータに使う同期式DフリップフロップのALS74です。通常なら，ほかのICと同じようにLSシリーズLS74を使えばよいはずなのですが，ここでは動作速度の速いALSシリーズでなければなりません。

その理由を図2のタイミングチャートを見ながら考えてみましょう。まず，

01＋01＝10

の演算を行っているものとします。これは，レジスタの出力(01)とスイッチからの入力(01)が，ALUの2つの入力に入っていると同時にその演算結果(10)がレジスタの入力に戻ってきていることになります。レジスタの出力にスイッチからの別のデータが加算されているわけですから，その演算結果であるレジスタの入力(10)は，そのときの出力の値(01)と異なっている場合がほとんどです。

そこで，この演算結果をレジスタに格納するためにクロック信号を送るとします。クロック信号が立ち上がった瞬間にレジスタにデータ(10)が取り込まれ，同時にそのデータが出力側に出てきます。ところが，レジスタの入力はALUの演算結果であるわけなので，レジスタの出力値が(10)に変わった瞬間に入力もまた（01＋10＝11）に変わってしまうことになります。いい換えれば，クロック信号が立ち上がった瞬間，ロックすべきデータ(10)が入力されているわけではなく，新しいデータ(11)が入力されていることになります。特に下位ビットについては，図2-2にもあるようにデータがクロックの立ち上がり時間内に0→1に変動してしまっています。

ここで，レジスタのホールド時間について思い出してください(図3)。ホールド時間というのは，クロックが立ち上がってから，どれだけの時間，入力データを一定にしておかなければならないか，というものでした。いまの場合，クロックが入った瞬間，すでに入力データが変わってしまうので，ホールド時間は0でなければならないということになります。もし，ホールド時間が0でなければ今回の設計ではアキュムレータに正しくデータが格納されないことになります。Dフリップフロップの規格を

図1　回路図

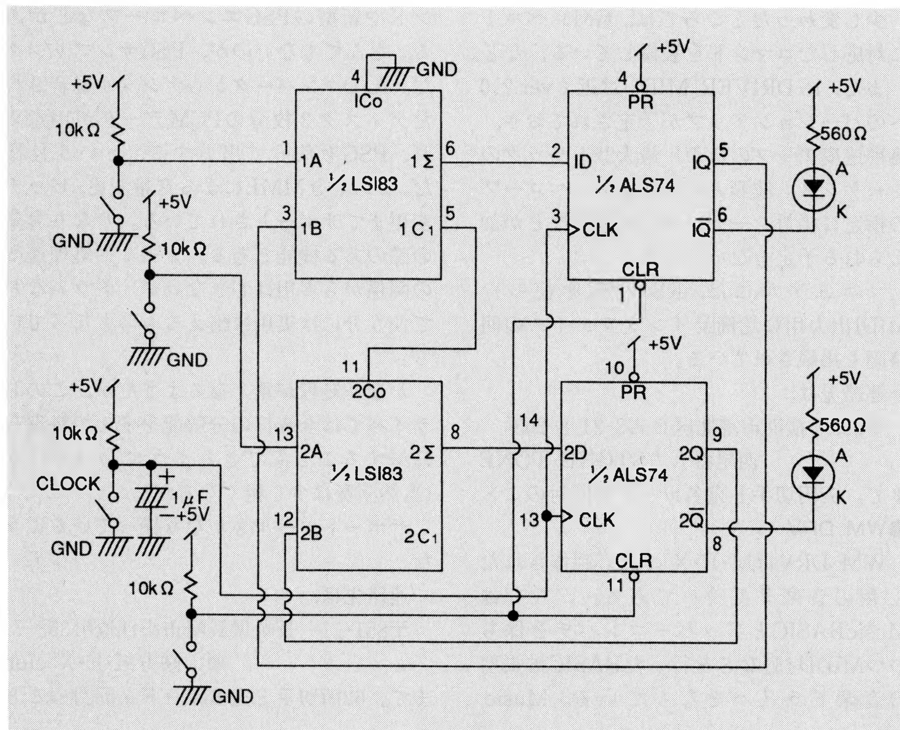



表1　部品表

| | | |
|---|---|---|
| 74LS183 | 1個 | 230円 |
| 74ALS74 | 1個 | 40円 |
| 2ビットDIPスイッチ | 1個 | 80円 |
| 押しボタンスイッチ | 2個 | @100円 |
| ICソケット14ピン | 2個 | @30円 |
| TLR113A | 2個 | @20円 |
| IC基板ICB-86（サンハヤト製） | 1枚 | 90円 |
| 10kΩ抵抗 | 4本 | @2円 |
| 560Ω抵抗 | 2本 | @2円 |
| 1μF電解コンデンサ | 1個 | 10円 |

図2 レジスタデータの変動

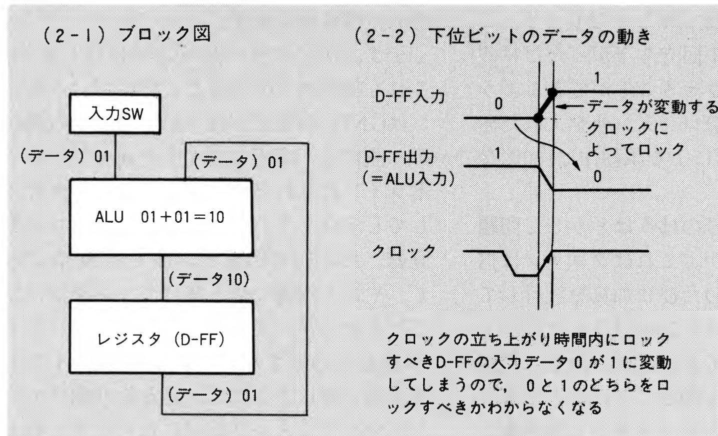



図3-1 ホールド時間

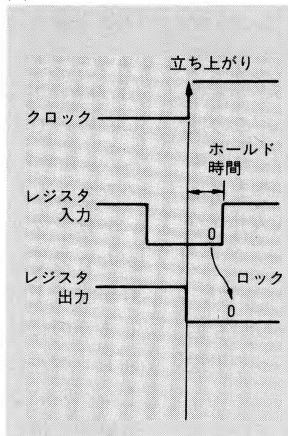
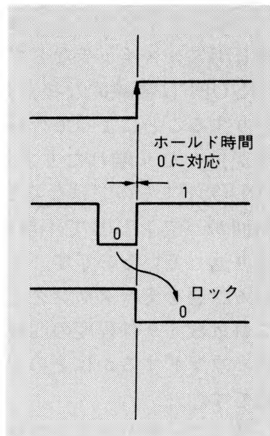
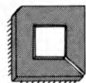



図3-2 いまの回路の場合




見るとLS74はホールド時間が5nsになっているので，うまくいかないことが考えられるのです。それに対して高速版のALS74ではホールド時間が0nsとなっており，今回の設計でも大丈夫なはずです。それで，今回はALS74を使うことにしたのです。



## 反転出力端子と表示LED

LEDについても前回と同じTLR113Aを使いました。前回は回路の都合で，LEDが点灯しているときが数字の0に，消灯しているときが1に対応していましたが，これでは混乱するので今回は，Dフリップフロップの反転出力端子$\overline{Q}$をうまく使うことによって点灯状態が1，消灯状態が0となるように工夫しました。

LEDは一般的に図4のような接続で使用しますが，このときに出力端子がHの場合だとLEDに電流が流れずに消灯状態になり，出力端子がLの場合にはLEDに電流が流れて点灯状態になります。H→1，L→0と対応させるだけなら，図のように出力端子にNOT回路を入れて出力を反転させなければなりません。

さて，今回使用したDフリップフロップALS74の規格を見ると，レジスタの出力であるQに対して，常に出力が反転している$\overline{Q}$があることがわかります。したがって，LEDを本来データが乗っているQのバスラインにつなぐのではなく，$\overline{Q}$に接続してやることによって，見掛け上バスラインの信号を直接表示しているように見せることができるのです。

今回は回路の都合で繰り上がりの3桁目の表示は行いませんでした。もしそこまでやるとしたら，もうひとつALS74を使って，LS183の2C端子の出力をほかの桁と同じようにクロック信号でロックすることになります。

図4 出力LED

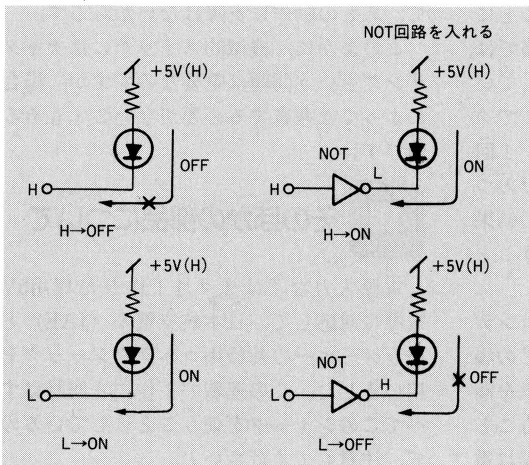
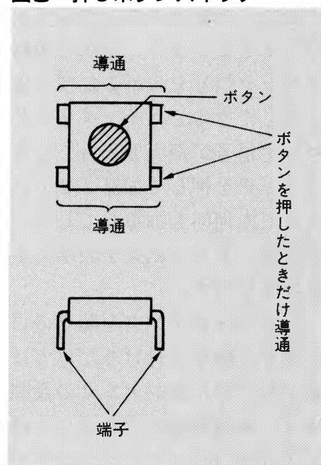
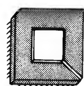



図5 押しボタンスイッチ



## スイッチとチャタリング

回路図左上の2個のスイッチは，データ入力用の2ビットDIPスイッチで，これは最初の加算器回路にも使われたものです。回路図では，2個別々のスイッチのように書いてありますが，実際はひとつの部品に小さなスイッチが横に2列並んでいて，それぞれのスイッチに対応する端子が2本ずつ計4本付いています。スイッチをONにするとその2端子間が導通し，OFFにすると断線します。今回の回路でも以前の加算器回路と同様に，スイッチをショートすると入力は0（Lレベル）になるように対応させていますので，基板に取り付けるときにはスイッチを下げるとON，上げるとOFFになるようにします。たいていのものはOFFという印が付いているので，それで確かめることができます。

このほかにクロックとクリア信号用の押しボタンスイッチが1個ずつ使用されています。これには互いに導通しているもの2つの端子が2組付いています(図5)。ボタンを押したときにそれぞれの組の間が導通し，離すとまた元に戻ります。ここで重要なのは，離したらバネでまた戻る「跳ね返り式」スイッチを使うという点です。

この跳ね返り式押しボタンスイッチを1回押すごとにクロック信号あるいはクリア信号を手動で出すわけですが，ここでこのような機械接点のスイッチで必ず注意しなければならない問題について説明したいと思います。

クロック信号のスイッチのところに入っている1μFの電解コンデンサがキーポイントです。試しにあらかじめこのコンデンサを取り付けないで動かしてみるとどうなるでしょうか。実際，私自身でこの回路を組んだときにこのコンデンサを取り付けずにテストしてみました。クリア信号は問題なかったので，今度は，LEDの表示が00になる(両方とも消灯)ようにしてから入力DIPスイッチに値をセットし，クロックボタンを押してみると，スイッチにセットしたとおりにデータがレジスタにロックされません。同じデータをセットして何度か繰り返してみると，そのつど動作が違ってめちゃくちゃなデータを表示します。これは「チャタリング」という問題のために正常に動

作していないのです。

では，チャタリングとはなんでしょうか。押しボタンスイッチなどでは，スイッチのON/OFFは機械的な接点が付いたり離れたりすることによって行われます。この接点が付いたり離れたりする瞬間に一度にON/OFFするのではなくて，図6のように何回かバウンドして不規則にON/OFFを繰り返しているのです。このバウンドしている状態をチャタリングというのですが，これは数ミリ秒程度の間続いて，しかも何回バウンドするかはそのときによって不規則です。

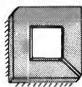
そして，チャタリングを起こしているときは，信号がH/Lを繰り返しているので，まさに不規則なクロック信号と同じことになってしまいます。今回の加算器回路では，クロック信号を入れるたびに値を足していってしまうので，チャタリングによってクロック信号が何度も入ってしまうと，1回スイッチを押して加算させているだけのつもりでも何回も加算してしまい，その結果めちゃくちゃな結果を表示してしまうことになるのです。

このチャタリングを防ぐのには，コンデンサを1個取り付けるだけでほとんどの場合十分です。コンデンサの役割は電気を溜めて，両端の電圧をしばらく維持することです。図7-1のようにコンデンサの両端に電池で一定電圧をかけておくと，コンデンサに電気が蓄えられ，電池を取り外したあともしばらくの間は電圧値を保ちます。電池を取り外した直後からのコンデンサの両端の電圧を，時間の関数としてグラフに表すと図7-2のようになります。

いま，図7-2のようにチャタリングが起こっている部分にコンデンサを付けると波形がなまされて，電圧が下がりきらずにON/OFFを繰り返す状態ではなくなります。このことによって，1回立ち下がっただけの信号のように動作させることができるようになるのです。実際コンデンサを入れたところ，チャタリングによる誤動作は100%なくなりました。

では，クリア信号のほうはどうして問題がないのでしょうか。これはクロックは信号が立ち上がるそのたびに加算を実行してしまうのに対し，クリアのほうは信号が1回Lレベルになってしまえば，あとは何回Lレベルになっても同じことだからです。最終的に押しボタンスイッチから手を離してしまえば，クリア端子はHレベルになって，あとの動作に支障はないからです。

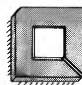
このように，機械的スイッチにはチャタリングという問題は重要なのですが，場合によっては考慮する必要がないこともあるのです。



## そのほかの部品について

電源入力端子は先々月工作した専用5V電源に対応して，日本航空電子（JAE）というメーカーの基板用コネクタジャックを用いました。この連載で工作する回路はすべてこのジャックを使うことにしているので，注意してください。

基板はサンハヤトのICB-86というIC専用基板で，横型のためTTL IC 2個分のスペースがあります。これまでの連載で毎回のように使われてきた基板のICB-87は縦型でTTL IC 1個分のスペースしかないので今回はこちらを使います。また，ICB-87と同じように，中心に+5VとGNDのラインが通っていて配線に便利なようになっています。



## 製作実習

では，さっそく製作に入りましょう。実体配線図は図8のとおりです。部品の配置はTTL ICのLS183とALS74とを横に並べることで，残りの配置はほぼ決まってしまいます。出力LEDは上半分，入力スイッチは下半分にまとめて配置しています。電源端子は横に通っている電源ラインのために基板の横に出します。

まず，ICソケットをハンダ付けします。これも毎回述べていることですが，7番ピンはGND，14番ピンは+5Vなので，内側に折り曲げて，基板の真ん中を通っている2本ラインにそれぞれいっしょにハンダ付けしてしまいます。LS183の4番ピンはGND直結，ALS74の10番ピンは+5V直結なので，やはり内側に折り曲げてハンダ付けしてしまいます。実はALS74の4番ピンも+5V直結なのですが，これは+5Vラインと反対側の列にあるので，足を折り曲げてハンダ付けするというわけにはいかず，あとからジャンパ線でつなぎます。そのほかのピンはそのままそれぞれのパターンにハンダ付けします。

次に電源端子を取り付けます。これはちょうど+5VとGNDラインにハンダ付けします。2ピンジャックは差し込むほうから見て右側が+5V，左側がGNDなので，向きを間違えないようにしてください。

今度はスイッチ群を取り付けましょう。2ビットDIPスイッチには，ON/OFFの向きがありスイッチを下にしたときにON（導通）になるように上下逆に取り付けます。スイッチ本体にOFFという表示が書かれており，それが上にくるようにします。スイッチは4本の足をすべて差し込んだ位置にハンダ付けしてください。

今度はこのスイッチにつながる抵抗を取り付けます。抵抗の足は，差し込んだら折り曲げて，スイッチの足のところまで延ばし，そこにいっしょにハンダ付けします。スイッチの下位ビットはLS183の1番ピンとつながるので，ちょうどそこにくる抵抗の足を利用していっしょにハンダ付けしてしまいます。反対側の足は2本まとめてハンダ付けし，+5Vに直結します。押しボタンスイッチにも向きがあるので，注意してください。

初めから導通している2組の足がそれぞれ横に並び，スイッチを押したときに上下の足が導通するように取り付けてください。そして，これら2個の押しボタンスイッチにつながる抵抗を取り付けます。ハンダ付けする面から見て右側の押しボタンスイッ

**図6 チャタリング**

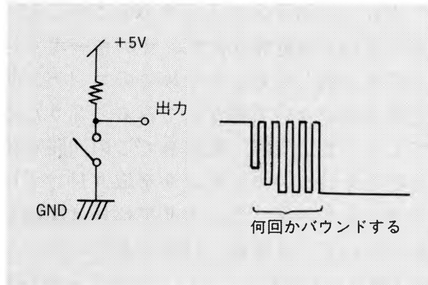



**図7-1 コンデンサ**

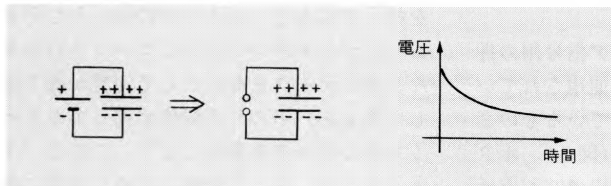
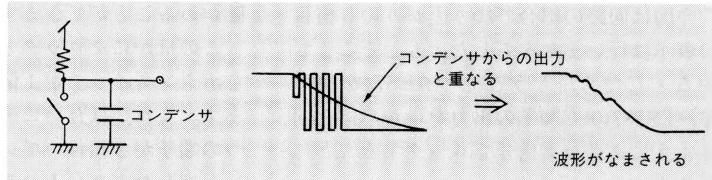



**図7-2 チャタリング防止回路**

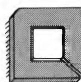


チがクリアスイッチで，ALS74の１番，13番ピンにつながります。ちょうど１番ピンのそばにくる抵抗の足を利用して，いっしょにハンダ付けしてしまいます。

左側の押しボタンスイッチがクロックスイッチで，ALS74の３番，11番ピンにつながります。これはちょうど３番ピンのそばにくるので，そこにくる抵抗の足を利用していっしょにハンダ付けしてしまいます。反対側の足は延ばして２本の抵抗といっしょにハンダ付けします。そこも＋５Ｖに直結です。

LEDを取り付けます。LEDのアノードとカソードの向きを間違えずに２個並べて差し込み，計４本の足を差し込んだ場所でハンダ付けします。最後にこのLEDにつなぐ抵抗器を取り付けます。この抵抗も差し込んだ足を折り曲げて，配線材の代わりに使います。LEDと反対側の足はそれぞれ＋５Ｖに直結します。

以上で部品の配置は終わりました。あとは，部品の間を配線材でつながなくてはなりません。ビニール被覆線で間を飛ばしてつなぐのが11本，スズメッキ線で基板に這わせてつなぐのが３本あります。特にスイッチ群の片方の端子をGNDに落とすために，基板のいちばん下にスズメッキ線を這わせます。実体配線図上の対応をよく確認しながら，ハンダ付けしていってください。



## 動作チェック

LS183とALS74とをソケットに差し込み，DIPスイッチを全部下にして電源をつないでみてください。なおTTL ICの向きは，ICを上から見ると切り欠きがあり，その切り欠きを基準に左回りに１～14番ピンになっています。この時点ではLEDは不規則に点灯し，特に意味はありません。そこでクリアスイッチを押した瞬間に２個のLEDが消灯したら，ひとまず成功です。

次にDIPスイッチの下位ビットだけ上に上げてデータ01を入力し，クロックスイッチを押して，離した瞬間に下位ビットのLEDだけ点灯（01の表示）したらまず問題なく動作しています。そのままでもう一度クロックスイッチを押すと，今度はLEDが10を表示します。このとき，

　01＋01＝10

を演算していることになります。同じようにもう一度クロックを押すと，

　10＋01＝11

もう一度押すと次は繰り上がりがないので，

　11＋01＝00

となります。あとは，任意の加算が実現できますので，試してみてください。

ではうまく動作しなかったときはどう対処するのがよいでしょうか。今回の回路はALUとアキュムレータの２つのブロックに分かれているので，それぞれチェックしていくという手があります。

LS183の部分については，単なる配線ミスによる誤動作以外まず考えられません。回路図あるいは実体配線図をしっかり追っていて間違いを見つけてください。LS183の出力端子1Σ，2Σ（６番，８番ピン）の電圧をテスターでモニタしながら，DIPスイッチを切り替えたときに出力が変化することをまず確かめてください。そして，各入力端子1A，2A，1B，2Bの入力データを正しく加算して出力していればOKです。

次にALS74のチェックポイントについて述べてみましょう。

1)　LEDが点灯したままか，まったく点灯しない

プリセット端子あるいはクリア端子周りの配線を確かめてください。まったくLEDの状態が変わらないとすれば，これらの配線が怪しいと考えられます。LEDの足の向きが逆の可能性もあります。



illustration : Y.Kawahara

2)　クロックスイッチを押すとLEDが切り替わるが，正しく演算しない

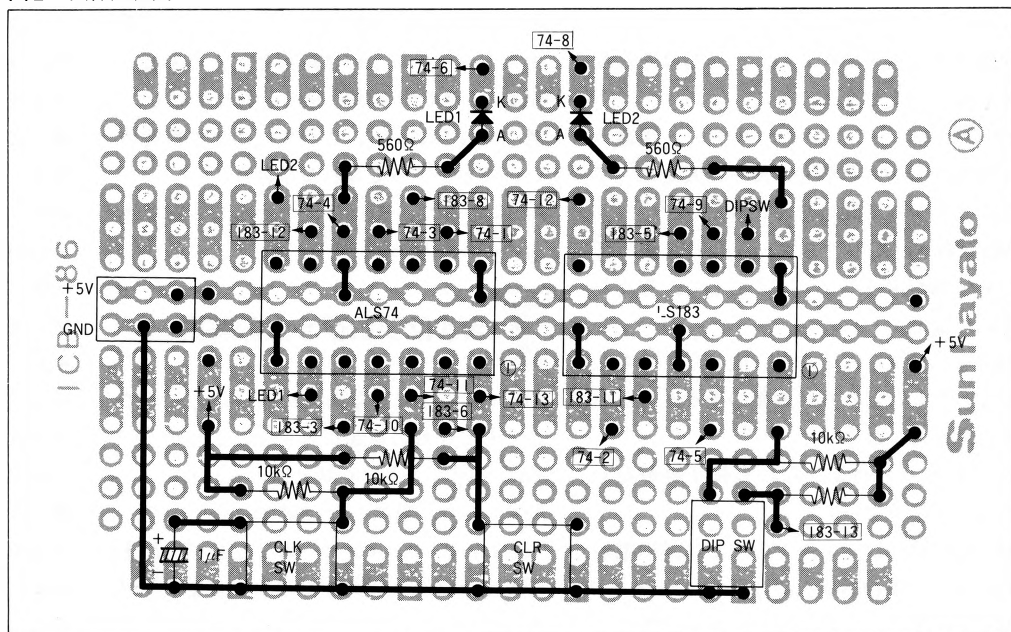
まず，桁がどこかで入れ違っているかもしれません。入力スイッチ，LS183，ALS74，出力LEDの桁の対応をしっかり確認してください。そこがOKなら，クロック端子周りの配線が怪しいと考えられます。どうしても配線ミスがなければ，チャタリング対策が十分でないので，コンデンサの値を10μFまで上げてみてください。あるいは，電解コンデンサの足の向きを間違えているかもしれません。

＊　　＊　　＊

今月でアキュムレータ付き２桁加算器が完成しました。これで，CPUで最も大切なALUとレジスタの考え方までマスターしてきたことになります。さらにX68000に近づくために必要となる回路は何でしょうか？

これはまた，来月のお楽しみです。

**図8　実体配線図**






マシン語カクテル in Z80's Bar
――――第38回

滑らかさとはなにか，別に長老の頭が滑らかになったという話ではありません。キャラクタの動きの話です。直線的な動きと曲線的な動き，どちらが滑らかかは直感的にわかりますが，この滑らかさはどこからくるのでしょうか。

# 憧れの導関数

Shibata Atsushi **柴田 淳**

**柴田淳**（以下Ats）：ねえねえ，除毛フォームってあるじゃないですか。

**ようこ**（以下Yo）：ムダ毛処理に使うムースのことね。

**Ats**：あれってよく考えると，結構危険ですよね。

**Yo**：危険って，含まれてる成分がってこと？

**Ats**：いやそうじゃなくて，たとえば視力の弱いお年寄りなんかが，たまたまオシャレ心を起こしたとするじゃないですか。

**Yo**：うん，それで？

**Ats**：いや，老人じゃなくても，ものすごく寝ぼけたおねーちゃんでもいいや。で，髪の毛整えようとして，スタイリングムースと間違って，除毛フォーム使っちゃうってことないんですかね。

**Yo**：さあ，ありそうな話だとは思うけど，聞いたことないわね，そういう人のこと。ねえ，マスターはどう？

**マスター**（以下M）：わ，わたしはそんなカン違いしませんよ。

**Yo**：なにいってんのよマスター。間違って除毛フォームを整髪に使った人を知ってるかどうかって聞いたんじゃない。

**Ats**：あっ，でもマスター，今日に限ってベレー帽かぶってマスクまでして，なんか怪しいなあ。

**M**：なにいってんですか，失礼な！　もともとわたしには，抜ける髪の毛なんてないんですから。ほらっ！

**Yo**：わかったわよ。なにも帽子脱がなくてもいいじゃない。

**M**：だいたい，ペンギンのくせに髪の毛があるほうが不自然なんだ。

**Ats**：ああっ，それってもしかして僕のこといってるんじゃ……。好きでこんなふうに描いてもらってるわけじゃないのに。

♪カラーン，コローン

**長老**（以下老）：ふぉっふぉっ，だいたいの話は扉の向こうで聞きましたぞ。ペンギンに髪の毛があるのは不自然とな。

**M**：誤解しないでください。別に長老のことをいったわけじゃないんですよ。

**Yo**：あれ，でも長老も帽子なんかかぶって，どうしちゃったんですか？

**老**：ふぉっふぉっ，それご開帳じゃ。

**Ats**：ああっ，長老スキンヘッド！

**Yo**：それって話の流れからいくともしかして……。

**老**：間違って除毛フォームを使ったのはわしじゃ。モウロクしたかのう，ふぉっふぉっふぉっ。

**M**：でもこれで，ペンギンとして不自然なのは柴田君だけになったわけですよね。

**Yo**：そうねえ，確かにひとりだけ髪の毛生えてるっていうのは統制を欠くわよね。

**Ats**：イヤですよ僕，髪の毛剃るなんて。

**老**：ふぉっふぉっ，この先いつまで威勢のいいことをいっておれるかのう。

**Ats**：あんた，またなにか企んでるんじゃないだろうなあ。



## まずはプログラムの解説

**Ats**：さて，適当な人数が揃ったところで，さっそく今日の本題に入りましょうか。

**Yo**：今回は工夫なく話を進めるわね。

**Ats**：まあ，気にしない気にしない。で，ですね，アクションゲームなんかで，たまにキャラクターの動きがギザギザなやつがあるでしょう。

**M**：最近はあまり見かけませんけどね。

**Ats**：そこでまず最初に，そのギザギザを取り除くことを考えてみましょうか。

**老**：そこで今回のプログラムが出てくるわけじゃな。

**Ats**：説明はあと回しにして，とりあえず走らせてみましょう。いちばん近いところにいるようこさんにお願いしようかな。

**Yo**：いいわよ。ええと，アセンブルしてJA000でいいのね。

**老**：ほほう，画面にギザギザの波が現れたのう。

**M**：こういうのをノコギリ波っていうんですよね。

**Ats**：ちなみにこの横棒は，42行からのBAR1っていうサブルーチンで動かしてます。で，今度は29行の#BAR1を#BAR2に変えて走らせてみましょうか。

**M**：なるほど，ノコギリ波がサイン波に変わった。

**Yo**：確かにこっちのほうが動きとしては滑らかね。メリハリがあるっていうか。

**老**：この棒の動きは76行からの#BAR2で管理しているわけじゃな。

**Ats**：では次にプログラムの中身を見て，2つの動きはどのように違うかを，もっと明確にしてみましょう。

**M**：まず#BAR1からですね。

**Ats**：ここでやっていることっていうのはものすごく単純で，横棒の高さを見て，いちばん上にきていたら棒を動かす方向を下に向けて。

**老**：いちばん下にきていたら上向きに動かすようにするというわけじゃな。

**Ats**：あ，そうだ。いい忘れてたけど，横棒にはそれぞれ4バイトずつのテーブルが

割り当てられています。

Yo：そのテーブルの内容が表１に書かれてるのね。

Ats：この表を頭において，今度は#BAR2を解析してみましょう。

M：リストの長さからみても#BAR2のほうは２倍あるから，さっきみたいに簡単にいきそうもないですね。

Ats：まあ２倍とはいっても，後半部分でやっていることというのは#BAR1と変わりません。119行から125行までを見てください。

老：本当じゃのう。ここでやっとるのは，まずテーブルの２番目の０ビットを見て，立っておったら横棒の高さを増やす，寝ておったら減らすということだけじゃ。

Yo：判定部分が削られている分，むしろ簡単になってるわね，#BAR1と比べて。

M：とすると問題は，その前の部分ということになりますね。

Ats：で，表を見ると，#BAR1で使ってなかったテーブルがありますよね。

Yo：テーブルの３番と４番ね。

Ats：#BAR2の前半部分では，主にこの２つに関する処理をしています。

老：具体的にはどのような処理をしておるのじゃろうか。

Ats：ええとですね，まず頭のところで，テーブルの２番目の第１ビットを見て処理を振り分けています。

Yo：振り分けるっていうと？

老：ビットの状態によって，テーブルの値を増やすか減らすかということじゃろう。

Ats：その増やすなり減らすなりした増分，つまりテーブルの３番目の値を，４番目のカウンタに足しているのが113行目です。

M：あの，ちょっと気づいたことがあるんですけどいいですか？

Ats：どうぞご遠慮なく。

M：このプログラムってもしかして，1991年の12月号でやったことと一緒なんじゃないですか？

Yo：そういえばそうね，あのときも似たようなことやったわよ。

表１　テーブルの内容

| | |
|---|---|
| 0 | 横棒の高さ |
| 1 | 方向を表すフラグ |
| 2 | 増分 |
| 3 | 増分用のカウンタ |

Ats：手っとり早くいっちゃえばそうなんですけどね。まあそのことはおいといて続きをやりましょう。

老：増分をカウンタに足してからどうするかじゃな。

Yo：あのときとやってることが同じなら，私にも説明できるわ。増分とカウンタを足し終わったら，足したあとのカウンタの値を調べるのよね。

Ats：調べた結果，ここでは32を超えていたら，カウンタから32を引いて。

Yo：次の横棒の高さを増減する処理を通るようになってるのよね。

老：もし32を超えていなかったら，なにもせずにおくのじゃな。

Ats：と，一応プログラムの説明は終わったわけですが。

M：ですがってことは，まだ先があるんですね。

Yo：これ以上どういう具合に話を続けるのかしらね。

Ats：まあ見ててくださいよ。



## 滑らかさのモト

Ats：突然ですが，滑らかさとはいったいなんでしょう。

M：ホントに突然きましたね。

Yo：えーと，滑らかってことは，ギザギザじゃないってことなんじゃないの？

老：ふぉっふぉっ，それでは漠然としすぎていて要領を得ないのう。

Ats：それじゃ質問のしかたを変えてみましょうか。「滑らかさ」を数学的に定義するとどうなるでしょう。

Yo：そういう質問を私にしないでよね。

Ats：あっ，ようこさんは開き直っちゃった。じゃあマスターはどうですか。

M：数学的に，ですか？　ええと，幾何学でいえば，たとえば１次関数より２次関数のほうが滑らかですよね。

老：つまり次数が高いほうが滑らかということになるかのう。

Ats：あっ，いいセンいってるなあ。そこまでわかってれば，あとは図を使ったほうが話が早いや。まず図１を見てください。

Yo：さっきのノコギリ波と階段みたいなのが縦に並んでるやつね。

Ats：これは上が#BAR1のサブルーチンで動かしたときの横棒の動きを，下がその増分をグラフ化したものなんですが，これを見てなにか思い出しませんか？

M：さあ，うーん，困ったなあ……。

Yo：なにも困ることはないじゃない，マスター。

老：おお，わしは昔バアさんと初めての接吻をした杉並木を思い出したぞ。

Ats：そういうことをいってんじゃないんですよ，まったく。じゃあ今度は図２を見てください。内容は図１と同じものです。

Yo：どう，マスター？

M：さあ，わたしはさっぱり。

老：おお，わしは戦前盗んだ，隣のご婦人の腰巻きの柄を……。

Yo：やだ長老，昔からそんな人だったの。

老：いやのう，そのご婦人というのが美人でのう，わしは盗んだ腰巻きを枕に……。

Ats：いったいなんの話してるんですかっ，人が一生懸命説明してるのにっ！

老：わかったわかった。要するになにがいいたいのじゃ。

M：あっ，そういえば図２の増分のグラフも，やっぱりノコギリ型してますよね。

Ats：今日のマスターはなんだかさえてる

図１　#BAR1での動き

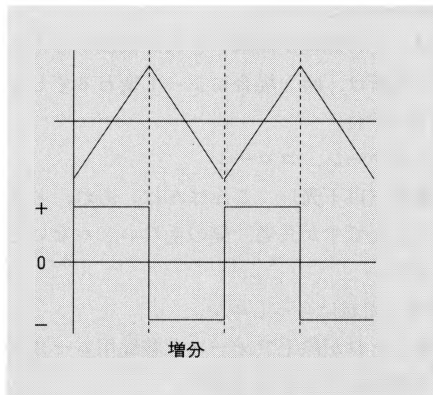
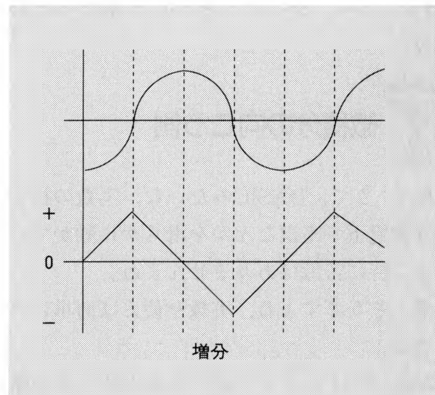



図２　#BAR2での動き






なあ。そうなんですよ，それがまずひとつ。
**老**：ということは，この4つのグラフというのは，実は3種類のグラフだといえるわけじゃな。
**Yo**：なるほど。図1の増減がいちばん下にきて，真ん中がノコギリ波で，上がサイン波ってなるわけね。
**Ats**：そこでですよ，その3つに並べ換えたグラフを上から下に見ていくとするじゃないですか。するとどうなっていきますか？
**老**：だんだんギザギザになっていくのう。
**M**：あれ，待ってくださいよ。さっきの話だと，より滑らかにするには関数の次数を上げればいいってことだったじゃないですか。ここでは滑らかでなくなっていくんですから，逆に次数は下がっているってことですよね。
**Ats**：ここまでたどり着けば，今回の話は90パーセント終わったも同じです。
**Yo**：わたしにはまだ話がよく見えないんだけど。
**Ats**：つまりですね，この図の増分っていうのは，上にあるグラフの導関数になってるんですよ。
**M**：つまり，微分したあとの関数ってわけだ。
**Yo**：ひー，やっぱりそういう話だったのね。



## 微積分の向こう側

**Ats**：さて，BASICみたいな，実数の扱える環境下で座標なんかを滑らかに動かすには，特に苦労はありませんよね。
**M**：そうですよね，実数を使えば簡単にできる。
**Ats**：だけどマシン語では基本的に正の整数しか扱えないから，そのために増分と増分用のカウンタを用意するわけですが。
**老**：その増分とか差分とかいうのは，よく見てみると1次導関数じゃったのじゃな。
**Ats**：つまりですね，デジタルな世界の中では，滑らかな動きとこの導関数っていうのは，かなり親密な関係があるんですよ。
**Yo**：その具体的な例として，今回のプログラムがあるってわけね。
**M**：ところでふと思ったんですけど，図2の曲線があるじゃないですか。それをもうひとつの増分とみなして，さらにその上に変数をとるとどんな曲線が描かれると思います？
**老**：そうじゃのう。その曲線は2次関数的なのじゃから，その上の曲線は，2次関数を積分したもの，つまり3次関数のようになるじゃろうな。
**Ats**：そうですね。さっきは次数が上がればそれだけ滑らかになるっていったけど，この場合はよりクネクネするだけでしょう。極値が増えるわけですから。
**M**：なるほどね。ということは，増分をたくさん取ったからって，必ずしも滑らかになるとはかぎらないんだ。
**Yo**：うーん，話の内容は半分ぐらいしかわからなかったけど，とにかく整数だけでもそれなりに滑らかさを出す方法があるってことだけはわかったわ。
**老**：それだけ理解できれば上等じゃろう。
**Ats**：で，詳しいことは知らないんではっきりしたことはいえないんですけど，この考え方って，特に画像処理で威力を発揮するはずなんですよ。
**Yo**：たとえば？
**Ats**：そうだなあ，低級なところだとラインを引くとか，スプライン曲線を描くとか。
**老**：また高級な部分じゃと，2次元画像の輪郭抽出とかのう。
**M**：どの処理が高級でどれが低級かっていう判断は，時と場合によって変わるでしょうけどね。
♪カラーン，コローン
**源光（以下光）**：こんばんは。あれ，どうしたんですか長老，髪の毛ないじゃないですか。
**Yo**：光君いらっしゃい。
**老**：それが除毛フォームと整髪用ムースとを間違えてしまってのう。
**光**：へぇ，長老も結構おっちょこちょいなところがあるんだなあ。で，マスターがマスクしてるのはなんですか？
**M**：あ，ああ，これはただの風邪です。
**光**：でも長老に髪の毛がなくなったら，高橋君描き分けができなくて困るんじゃないかなあ。たとえば，僕がこう，帽子を取ったりしたら……。
**Yo**：あっ，それ面白そうね。わたしもリボン取って，まつ毛切っちゃおっと。
**Ats**：ちょっとようこさん，そんなことして知りませんからね。
**Yo**：大丈夫よ。きっと来月にはもとどおりになるから。柴田君も思いっきり，髪の毛剃っちゃいなさいよ。
**Ats**：いやですよ。この髪の毛がなくなったら，僕というペンギンのアイデンティティは……。ああっ，光君なにするんだっ，そして長老の持っている白い泡の乗ったクシはいったい……。
**老**：ふぉっふぉっ，いうまでもないじゃろ。
**Ats**：ううっ，ちくしょう，こうなりゃヤケだ。さあみんな，横一列に並ぶんだっ！
**光**：オウ！
**M**：おう！
**老**：をう！
**Yo**：いいわよっ！
**Ats**：ワッハッハッ，前にも似たような展開があったような気がするけど，そんなこと気にするなっ！　どうだ高橋，オレたちを描き分けてみろ！
**光**：隊長！　イエローマスターがひとりだけマスクしてます！
**M**：イ，イエローマスターって……。
**Ats**：口答えするなイエロー！　さっさとマスクを取るんだ！
**M**：いやですよ，ただでさえこんなことバカバカしいんだから。
**Ats**：うるしゃい！　こうなりゃ力ずくだ，それっ！
**M**：ああっ，やめてくださいよっ！
**Yo**：あれ，どうしたのマスター，口の周りカサブタだらけじゃない。
**M**：ううっ，今朝ひげ剃りと間違えて，毛玉取りでヒゲ剃っちゃったんですよ。
**光**：それを隠すためにマスクしてたんですね。
**老**：おあとがよろしいようで。

―つづく―

```
0000                ; ############
0000                ; # W A V E #
0000                ; ############
0000                ; for Z80's Bar
0000                ;
0000                ;   1993.FEB
A000                  ORG    $A000        ;START ADDRESS
1FF4 P              @PRINT  EQU  $1FF4
201E P              @LOC    EQU  $201E
2030 P              @WIDTH  EQU  $2030
1FD0 P              @GETKEY EQU  $1FD0
A000                ;
A800 P              #DATOP  EQU  $A800    ;DATA TOP ADDRESS
A000                ; SCREEN INIT
A000 3E 28            LD     A,40
A002 CD 30 20         CALL   @WIDTH
A005 3E 0C            LD     A,$0C
A007 CD F4 1F         CALL   @PRINT
A00A                ;
A00A CD 1F A1         CALL   #INIT
A00D 0E 02            LD     C,2
A00F 11 04 00         LD     DE,4
A012                #LOOP0
A012 DD 21 00 A8      LD     IX,#DATOP
A016 41               LD     B,C
A017 CB 28            SRA    B
A019                #LOOP1
A019 78               LD     A,B
A01A CD 33 A0         CALL   #BAR1
A01D DD 19            ADD    IX,DE
A01F 10 F8            DJNZ   #LOOP1
A021 0C               INC    C
A022 3E 50            LD     A,80
A024 B9               CP     C
A025 C2 2A A0         JP     NZ,#STEP
A028 0E 4F            LD     C,79
A02A                #STEP
A02A CD D0 1F         CALL   @GETKEY
A02D FE 45            CP     "E"
A02F C2 12 A0         JP     NZ,#LOOP0
A032 C9               RET
A033                #BAR1
A033                ; BAR MOVE
A033                ; IX=ATTR TOP
A033                ; A=BAR NUM
A033 E5               PUSH   HL
A034 D5               PUSH   DE
A035 C5               PUSH   BC
A036 6F               LD     L,A
A037 DD 66 00         LD     H,(IX+0)
A03A CD FD A0         CALL   #BRCLR
A03D DD 7E 01         LD     A,(IX+1)
A040 E6 01            AND    1
A042 C2 58 A0         JP     NZ,#DOWN1
A045 DD 7E 00         LD     A,(IX+0)
A048 3C               INC    A
A049 DD 77 00         LD     (IX+0),A
A04C FE 2C            CP     44
A04E C2 68 A0         JP     NZ,#STEPB1
A051 DD 36 01 01      LD     (IX+1),1
A055 C3 68 A0         JP     #STEPB1
A058                #DOWN1
A058 DD 7E 00         LD     A,(IX+0)
A05B 3D               DEC    A
A05C DD 77 00         LD     (IX+0),A
A05F FE 00            CP     0
A061 C2 68 A0         JP     NZ,#STEPB1
A064 DD 36 01 00      LD     (IX+1),0
A068                #STEPB1
A068 DD 66 00         LD     H,(IX+0)
A06B CD DD A0         CALL   #BRPRT
A06E C1               POP    BC
A06F D1               POP    DE
A070 E1               POP    HL
A071 C9               RET
A072                #BAR2
A072                ; BAR MOVE2
A072                ; IX=ATTR TOP
A072                ; A=BAR NUM
A072 E5               PUSH   HL
A073 D5               PUSH   DE
A074 C5               PUSH   BC
A075 6F               LD     L,A
A076 DD 66 00         LD     H,(IX+0)
A079 CD FD A0         CALL   #BRCLR
A07C DD 7E 01         LD     A,(IX+1)
A07F E6 02            AND    2
A081 C2 9E A0         JP     NZ,#DEC
A084 DD 7E 02         LD     A,(IX+2)
A087 3C               INC    A
A088 3C               INC    A
A089 FE 42            CP     66
A08B C2 98 A0         JP     NZ,#STEPINC
A08E DD 7E 01         LD     A,(IX+1)
A091 F6 02            OR     2
A093 DD 77 01         LD     (IX+1),A
A096 3E 40            LD     A,64
A098                #STEPINC
A098 DD 77 02         LD     (IX+2),A
A09B C3 B5 A0         JP     #STEPB2
A09E                #DEC
A09E DD 7E 02         LD     A,(IX+2)
A0A1 D6 02            SUB    2
A0A3 D2 B2 A0         JP     NC,#STEPDEC
A0A6 DD 7E 01         LD     A,(IX+1)
A0A9 E6 01            AND    1
A0AB EE 01            XOR    1
A0AD DD 77 01         LD     (IX+1),A
A0B0 3E 00            LD     A,0
A0B2                #STEPDEC
A0B2 DD 77 02         LD     (IX+2),A
A0B5                #STEPB2
A0B5 DD 86 03         ADD    A,(IX+3)
A0B8 4F               LD     C,A
A0B9 E6 3F            AND    63
A0BB DD 77 03         LD     (IX+3),A
A0BE B9               CP     C
A0BF CA D3 A0         JP     Z,#STEPNON
A0C2 DD 7E 01         LD     A,(IX+1)
A0C5 E6 01            AND    1
A0C7 C2 D0 A0         JP     NZ,#DOWN2
A0CA DD 34 00         INC    (IX+0)
A0CD C3 D3 A0         JP     #STEPNON
A0D0                #DOWN2
A0D0 DD 35 00         DEC    (IX+0)
A0D3                #STEPNON
A0D3 DD 66 00         LD     H,(IX+0)
A0D6 CD DD A0         CALL   #BRPRT
A0D9 C1               POP    BC
A0DA D1               POP    DE
A0DB E1               POP    HL
A0DC C9               RET
A0DD                #BRPRT
A0DD                ; BAR PRINT
A0DD                ; L=BAR NUM,H=HIGH
A0DD                ;
A0DD E5               PUSH   HL
A0DE D5               PUSH   DE
A0DF C5               PUSH   BC
A0E0 7C               LD     A,H
A0E1 CD 11 A1         CALL   #DEV
A0E4 67               LD     H,A
A0E5 3E 13            LD     A,19
A0E7 94               SUB    H
A0E8 67               LD     H,A
A0E9 CD 1E 20         CALL   @LOC
A0EC 11 FA A0         LD     DE,#BRTAB
A0EF 79               LD     A,C
A0F0 83               ADD    A,E
A0F1 5F               LD     E,A
A0F2 1A               LD     A,(DE)
A0F3 CD F4 1F         CALL   @PRINT
A0F6 C1               POP    BC
A0F7 D1               POP    DE
A0F8 E1               POP    HL
A0F9 C9               RET
A0FA                #BRTAB
A0FA 5F 2D 7E         DB     "_-~"
A0FD                #BRCLR
A0FD                ; BAR CLEAR
A0FD E5               PUSH   HL
A0FE 7C               LD     A,H
A0FF CD 11 A1         CALL   #DEV
A102 67               LD     H,A
A103 3E 13            LD     A,19
A105 94               SUB    H
A106 67               LD     H,A
A107 CD 1E 20         CALL   @LOC
A10A 3E 20            LD     A," "
A10C CD F4 1F         CALL   @PRINT
A10F E1               POP    HL
A110 C9               RET
A111                #DEV
A111                ; CALC A/3
A111                ; A=ANSWER,C=MOD
A111 D5               PUSH   DE
A112 06 03            LD     B,3
A114 16 FF            LD     D,255
A116                #LOOPDV
A116 4F               LD     C,A
A117 14               INC    D
A118 90               SUB    B
A119 D2 16 A1         JP     NC,#LOOPDV
A11C 7A               LD     A,D
A11D D1               POP    DE
A11E C9               RET
A11F                #INIT
A11F                ; DATA AREA INIT
A11F E5               PUSH   HL
A120 D5               PUSH   DE
A121 C5               PUSH   BC
A122 06 28            LD     B,40
A124 21 00 A8         LD     HL,#DATOP
A127                #LOOPIT
A127 36 17            LD     (HL),23
A129 23               INC    HL
A12A 36 02            LD     (HL),2
A12C 23               INC    HL
A12D 36 40            LD     (HL),64
A12F 23               INC    HL
A130 36 00            LD     (HL),0
A132 23               INC    HL
A133 10 F2            DJNZ   #LOOPIT
A135 C1               POP    BC
A136 D1               POP    DE
A137 E1               POP    HL
A138 C9               RET
```

速報

XCはもういらない？

# X68k Programming Series

Nakamori Akira 中森　章

**ソフトバンク書籍編集部の下にX68000用フリーウェア開発ツール作者が一堂に集まり，未曾有のプロジェクトを開始した。断片的であったツールがSXツール開発対応システムとして体系的にまとめあげられた。その概要をひと足お先に紹介しよう。**

## フリーソフトの威力

X68000はこれまでメーカーよりもユーザーによって支えられてきたマシンといえます。メーカーからはひと通りの開発ツールが供給されていましたが，それらは性能が悪い，処理速度が遅いといった批判があり，決してユーザーの満足できる代物ではありませんでした。X68000ユーザーの凄いところは，これらの不満をメーカーにぶつけて（そのうち行われるであろう）対処を期待するのではなく，自らがプログラムを改良してしまう点です。

今日ではパソコン通信が発達していますから，改良されたツールはフリーソフトとしてネットワークにアップロードされ，すぐにいろいろなユーザーの手に渡ります。そして，ユーザーからのフィードバックを受けてそのツールはさらなる改良が繰り返し続けられていきます。フリーソフトは，多くの場合はパソコン通信を通じて汲み上げられるユーザーの要求を最大限に取り込みつつ改良されていくのです。

X68000のフリーソフトの状況で最近（でもないけど）のトピックスはSX-WINDOWへの対応が挙げられます。ご存じのように，SX-WINDOWのプログラムはリエントラントな構造が要求（あるいは推奨）されています。従来はC言語のプログラムの記述を工夫することでコンパイル結果がリエントラントになるようにしていました。しかし，これはいかにも強引という感があり，C言語のプログラムとしては不自然さが残ります。

そこで，普通に書いたC言語のプログラムをコンパイラの側でリエントラントになるようにコード生成をしようという動きが起こりました。この問題はコンパイラだけでは対処できずアセンブラやリンカを巻き込んだ運動になり，結果としてSX-WINDOW用のリエントラントなコードを生成する機能がCコンパイラに備わりました。コンパイラが出力するコードだけではプログラムは動きません。そこでXC最後の牙城であったライブラリもリエントラントなものに書き直されることになったのです。これはまさに快挙といえるでしょう。

このようにフリーソフトには小回りの利いた開発体制において，メーカーの追随を許さぬものがあります。そして，そのような便利なツール群のうちで基本的な開発ツールである，Cコンパイラ，アセンブラ，リンカ，デバッガはもっとも多くの人の注目を浴びて改良され続けてきたものといえます。そして，今回これらの開発ツールがX68k Programming Seriesとしてまとめられることになったわけですが，まず最初にこれらの開発ツールがどのようなものか簡単に紹介しておくことにしましょう。

**●GCC（Cコンパイラ）**

GCCはGNUプロジェクトのCコンパイラで，ANSIに準拠しながらも独自の拡張を加えた最適化コンパイラです。それにX68000固有の機能を追加して移植したものです。GCCでコンパイルすると（控えめに見積もっても）XCの2倍以上の性能が出るようです。GCCは初期の段階ではいろいろな人が移植を試みましたが，現在は“真里子版GCC”と呼ばれる版がもっとも有名です。

真里子版GCCに見られるX68000固有の拡張には主に次のようなものがあります。

・2進数表現の追加（XCと互換）
・日本語識別子
・割り込み処理の記述が可能
・ダンプコンパイル
・疑似統合環境
・プロファイラ
・SX-WINDOW用コード生成
・GDBまたはSCD.X用コード生成

**●HAS（アセンブラ）**

HASはメーカー純正のAS.Xと上位コンパチのアセンブラです。アセンブル処理をすべてメモリ上で行うので非常に高速です。AS.Xにリネームすればそのまま高速版AS.Xとして使用できます。

AS.Xに比べ次のような機能拡張が行われています。

・ローカルラベルの使用可
・相対セクションに対応（SX-WINDOW）
・マクロ機能の強化

**●HLK（リンカ）**

HLKはメーカー純正のLK.Xと上位コンパチのアセンブラです。リンク処理をすべてメモリ上で行うので非常に高速です。LK.Xにリネームすればそのまま高速版LK.Xとして使用できます。

LK.Xに比べ次のような機能拡張が行われています。

・相対セクションに対応（SX-WINDOW）
・G++のオブジェクトに対応

**●GDB（デバッガ）**

GDBはGNUプロジェクトのデバッガで，C言語のソースコードレベルのデバッグを可能にします。それに独自の拡張を加えてX68000に移植したものです。

シンボリックデバッグ情報はメーカー純正のSCD.Xと同一のようです。

X68000独自の機能としては次のようなものがあります。

・相対セクションに対応（SX-WINDOW）
・画面切り替えによる単一モニタ環境

## X68k Programming Series の意義

X68k Programming SeriesはX68000用に配布されているフリーソフトのうち基本的な開発ツールを集めたものです。具体的には上で示したGCC，HAS，HLK，GDBを集めています。まず開発ツールとそのドキュメントを収録した#1 Develop.が発売になり，その後，各種ライブラリを収録した#2 libcが発売されます。

なんだフリーソフトを集めただけかと早合点をしてはいけません。確かに，これがパソコン通信をしていないユーザーにX68000のツールを供給するという目的なら少々安易な企画です。しかし，X68k Programming Seriesは少なくとも2つの意味で画期的な企画になっています。

第1はドキュメントの整備です。フリーソフトはドキュメント（マニュアル）とオブジェクトが別配布であることも多く，しかも，たび重なる機能拡張のために最新の機能を盛り込んだ完全なドキュメントが存在しない場合もあります。今回収録される

GCC，HAS，HLK，GDBに関しては機能拡張の面ではフィックスした感があり，この時点でドキュメントを整備し文書化することはユーザーにとって非常にありがたいものになるでしょう。

特に今回の執筆は移植者あるいは開発者自身の手によるものですから，まさしく決定版のドキュメントになると予想されます。

第2はフリーなライブラリの供給です。従来，GCCを利用するためにはXC(C Compiler PRO-68K）を持っていることが必須とされていました。これはなにもメーカーの顔を立てているわけではなく，GCCでのコンパイルにはXCに付属するライブラリが必要だったのです。このため多くのプログラミング開発はXCという市販品のライブラリに依存することになります。

このようにして作られたプログラムをフリーソフトなどの形態で配布することは厳密な意味では著作権や使用料の問題を引き起こしかねません。XCに関してはドキュメントに使用した旨を明記すればよいことになっているようですが，いまいちすっきりとしません。しかし，今回フリーソフトのライブラリが供給されることによって（著作権自体は作者が所有しているとはいえ），ここらへんの事情を気にしなくてよくなることは間違いありません。そして，これからC言語を始めようとしているユーザーにとってはわざわざXCを買う必要がなくなるというのも大きな魅力です。

## X68k Programming Seriesの内容

本書はGCC，HAS，HLK，GDBというX68000用にフリーソフトとして配布されている開発ツールのドキュメントです。「Vol.1 Programmer's Guide」，「Vol.2 Reference」の2冊構成になっていて，Vol.1では各ツールの特徴や機能の説明，Vol.2ではオプションスイッチの詳細とエラーメッセージの紹介になっています。

表1に具体的な章立てを紹介しましょう。これを見て感じるように，全体的にGCC，HAS，HLKの章とGDBの章で内容が大きく分かれているようです。Vol.1では，前者が特徴や機能の説明に重点が置かれているのに対し，後者は具体的な使用例を中心とした入門書のようになっています。

Vol.2では，前者がオプションスイッチの詳細説明とエラーメッセージの解説および対処法を記述してあり，後者はコマンドのリファレンスになっています。この点，内容が統一的でない印象を受けます。「Programmer's Guide」と「Reference」という観点から見ると後者（GDB）のほうが主旨に沿ったまとめ方になっているのではないでしょうか。しかし，片やプログラムを作るためのツール，片や作ったプログラムを検証するためのツールですから切り口が違ってくるのもしかたないのでしょう(筆者もそれぞれ異なるようですし)。

それはともかく，本書は各ツールのマニュアルという意味あいが濃いできになっています。これまで不明確だったツールの機能がすべて（なのかなあ？）網羅されているので，ある程度の力を持ったユーザーには非常に有用でしょう。ただ，初心者がそれぞれのツールの入門用として読むには少し不親切かもしれません。GCC，HAS，HLKに関しては，もっと入門書的な記述があってもよかったのではと思います。

なお，Vol.2の「GCC診断メッセージ」はGCCのすべてのエラーと警告がどんな場合に発生するかを示してあって圧倒されたことを付け加えておきましょう。

## C Compiler PRO-68Kとの関係

X68k Programming Seriesが完結すると，これまでGCCによるプログラミングで必要であったXC（C Compiler PRO-68K）のライブラリが不要になります。XCの立場はどうなるのでしょう。

X68k Programming Seriesの広告では「#1 Develop.はXCもライブラリも使用できる」，「#2 libcはXCのライブラリとしても使用できる」という記述があり，お互いに共存しそうな意味あいを漂わせています。しかし，これが意味のない記述であることは誰の目にも明らかです。X68k Programming Seriesが完結するとXCは無用の長物と化してしまうでしょう。

XCにはかなり冷たいことを書いてしまいましたが，XCにはまだ存在価値が残されています。それはBASICコンパイラとしての生き方です。Cコンパイラが優勢になったとはいえ，X-BASICのユーザーはまだまだ多数派です。X-BASICのプログラムを高速化する手段としてのXC（本当はBC.Xがあればよいのですが）はまだこれからも現役であり続けるでしょう。

*　*　*

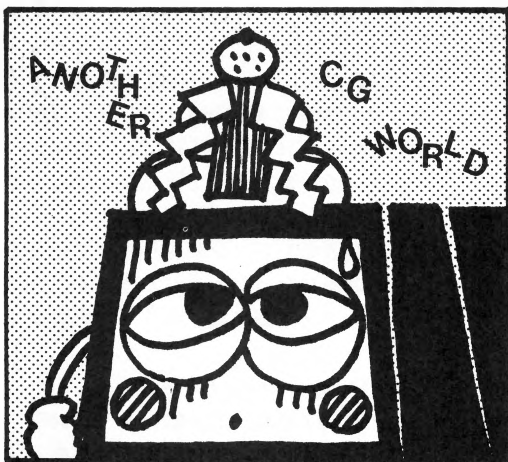
X68k Programming Seriesはかなり攻撃的な企画であるといえます。これはメーカーが開発ツールの改良に消極的であることへのユーザーの不満の発露といってよいかもしれません（FLOAT2.XやFLOAT3.Xの改良で希にメーカーの底力を見せてくれるのですが)。GCCやGDBなどの移植物はともかく，HASやHLKなどの基本ツールはメーカーが率先して改良すべきものでしょう。

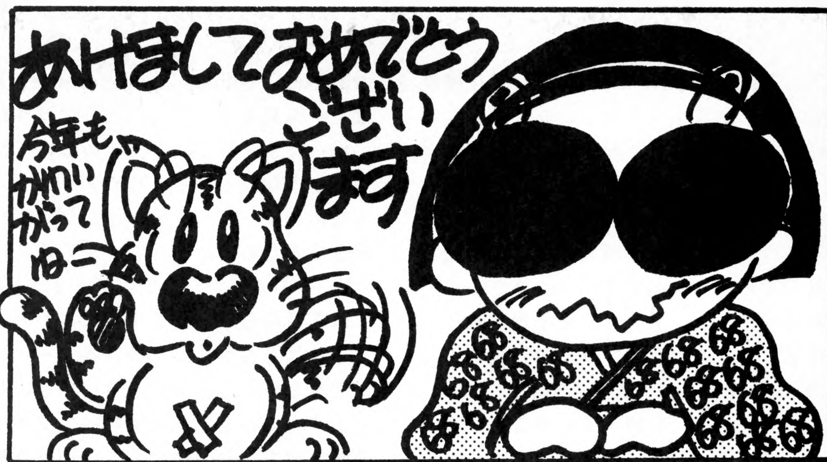
この発表でハドソンやシャープにとってはよい刺激になったかもしれません。今後のソフトウェア製品にどのような影響が出てくるか注目したいところです。

X68k Programming Series#1
B5判2冊組　5"2HD付き 9,800円
1993年1月発売予定
ソフトバンク　☎03(5488)1360

表1 X68k Programming Seriesの章立て

《Vol.1》
●Chapter 1　X68000 GCC
1.1 …GCCが扱うファイル
1.2 …コマンドラインオプション
1.3 …GCCが使う環境変数
1.4 …GCC拡張機能
1.5 …SX-WINDOW
1.6 …一般的な拡張
1.7 …ROM化について
1.8 …GCCの非互換性
1.9 …X68000 GCCでのプログラム
1.10…バグについて
1.11…GCCの制限
●Chapter 2　X68000 HAS
2.1 …HASの概要
2.2 …アセンブリ言語の文法
2.3 …セクションとモジュール化機能
2.4 …マクロ機能
2.5 …アセンブラ疑似命令
●Chapter 3　X68000 HLK
3.1 …HLKの概要
3.2 …使用例
3.3 …HLKとLKの違い
3.4 …トラブルシューティング
3.5 …オブジェクトファイルから実行ファイルができるまで
●Chapter 4　GDB
4.1 …GDB
4.2 …とりあえずGDBを使ってみる
4.3 …GDBの起動と終了
4.4 …プログラムの実行を制御する
4.6 …データを調べる/修正する
4.7 …ブレーク/ウォッチポイント
4.8 …スタックの調査
4.9 …ソースファイル
4.10…シンボルテーブルを調査する
4.11…マシンレベルのデバッグ
4.12…コマンドシーケンス
4.13…カスタマイズ
4.14…Human68k版GDBの仕様制限
●Chapter 5　Appendix
5.1 …各ファイルのフォーマット
5.2 …シンボリックデバッグ情報
5.3 …ソースレベルデバッガの仕組み
《Vol.2》
●Chapter 1　オプションスイッチ
1.1 …GCCオプション
1.2 …アセンブラオプション
1.3 …リンカオプション
1.4 …デバッガオプション
●Chapter 2　メッセージの意味
2.1 …GCC診断メッセージ
2.2 …アセンブラのメッセージ
2.3 …HLKのメッセージ
●Chapter 3　GDBのコマンド
3.1 …コマンドリファレンス
3.2 …行編集をサポートするキー


ANOTHER CG WORLD


あけましておめでとうございます
今年もかわいいがってねー


お正月までおおいそがしー
んだんだ
なにやってんのー 手伝おうかー
ネコの手


仮想空間のおおソージよ!
おおいそがしー
ワイヤーフレームじょうたい


ふすまや障子のはりかえ…
うるとらまん


いすの生地はりかえ…


かべがみのはりかえ…

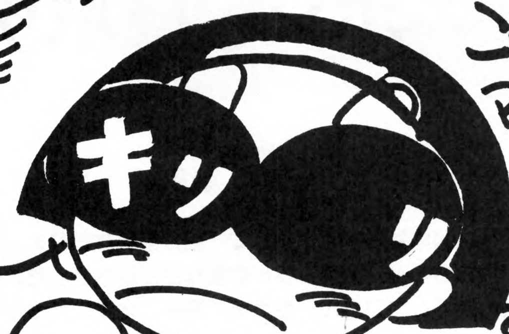

のり、ハサミは使わず、マッピングではりかえるざんす…
うつくしー
かあー
フニクリフニクラ♪
窓ごしの風景をエジプトのピラミッドからスイスアルプスにかえるとか…
生活感ただよう仮想空間だなあ〜







ボールとかの
まあるいもんの
ときはどーすんの？




こーゆー
マッピング
データ
用意
すんの
地球ギ貼り
やらなかったーて、
小学校でー

へーそーなのー




今回のCGデータ…
総物体数 601
光源 12
1280×1024ピクセル
1670万色 フルカラーを
4×5ポジで出力
使用ソフトは
C-TRACE、サイクロン
んで、山のマッピング・データは、
Oh!X編集部Aさんが
AMIGA2000+68030
アクセラレーターボードと
フラクタルソフトVISTA-PRO
で作ってくれました…
感謝！ いつも原稿
おくれてすんま
へーん






つーのは
ウン！
コレで
いーのよー

あーCGって
なんてお手軽



あたし向き
だねー
まったく
現実も
こーだといーのにねー
さてと、おソージおソージ

第76回

# 猫とコンピュータ

# ニボシを横取り

IDCソフトの中西秀樹さんから，教育ソフトの試作品が届けられた。すでに商品化されていたもののバージョンアップ版で，「感想を聞かせてください」ということなのだが，頼まれているのはもちろん私ではない。

高校生対象の教育ソフトで，内容は3単元あり，化学から「モル」，物理から「質点」，数学から「集合」だった。正式に発売する前にまわりの人の意見をなるべくたくさん聞きたいのだろう。

わが家では，化学はとくに夫の専門分野だし，トオルは高校生。感想をもとめられているのはこの2人にちがいない。「お子さんにもぜひ使ってみていただきたい」との一文もあった。

## ゲンソ・ゲンシ・ゲンシカク

「93年の1月1日に発売予定ですってよ。早く見なくちゃ」

と私は2人をせかす。だいじな製品を送っていただいたわけだし，「感想を……」なんて言ってもらえるのもありがたいことだ。なのに2人とも自分のしごとが先行して，なかなかソフトを実行してみるまでにならない。きちんと時間をとって，ゆっくり内容を見ようというつもりなのだろうけれど。

「じゃあ，私が先に拝見しまぁす」

化学も数学も劣等生だったのに，自分あてに届いたものではない学習ソフトが気にかかる。ニボシなんかきらいなホンニャアが，友だちのミミのニボシを気にするようなものだろうか。

ディスクの入った厚紙のパッケージの裏には，「登校拒否の子供に役立てばと思っています」と，ボールペンの走り書きがある。中西さんのお手紙には，ときどき封書の裏にこういうツブヤキの数行があって，心をひかれる。

ディスクをセットしてリセットボタンを押せばスタート。オープニングではウイルスチェックもおこなわれる。数秒で「異常は検出されませんでした」と報告が出てメニューがあらわれる。

「化学の尺度『モル』」を選択する。

「モル」の意味を理解することが，化学とは何か，化学にどう接したらよいかを知る基本的な手がかりになる，とこの単元の目標がメッセージで示される。

そして8つの項目がならぶ。「原子の大きさ」「原子の重さ」「原子核」「元素の原子量」「イオンと分子」……。

「原子の大きさ」では，原子とは身のまわりのものとくらべたら，およそどのくらいの大きさのものなのか，まず，この把握からはじまる。原子の直径と原子核の直径の比率はどれくらいなのか。これらをアニメーションをまじえて説明し，問いに対しての答えを入力させながらすすめていく。

画面は簡潔で見やすいけれど，簡潔なだけに，高校生でない私は用語を忘れてしまっているから，なかなか概念をつかみかねる。原子と原子核，電子の関係を，太陽系を例にして説明してあった。原子は太陽系全体，原子核は太陽，電子は太陽系の惑星たち，水星，金星，地球など。コメントをバックに，黄色の惑星がアニメーションで動きまわる。

なぁるほどと思うけれど，原子も電子も急に思い出した古い友人の名前のようで心もとない。そもそも，元素と原子ってどうちがうんだっけ。原子と陽子と電子の関係はどうだったかな。

学習ソフトはすべてを教えてくれるものではない。うまく道順をつくって理解の手助けをしてくれるのだ。よし，やっぱり復習しなくちゃ，ソフトに取り残される。

「トオル君，化学の教科書かして」

どうも1年生のときの教科書がいいらしい。そこで一気に化学の初歩に突入した。

## 動く教材

動いて，はたらきかけてくる教材が「学習ソフト」なら，子供時代，学校以外での「ソフト」といえば新宿のおばあちゃんだった。はじめから「おばあちゃん」ではないから，そのころは母だ。

このソフトは種類も豊富で量も多かったのに，私はじょうずに利用できなかった。相手のほうでも，私の力をひきだす工夫をしていなかったせいだろうと，いまでは思う。

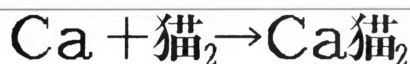
母は「私は勉強が好きで好きでたまらない」と，いまでもいう。だから小学校の教員になったのかというと，それはちょっと別なのだけれど。

とくに理科が好きで，かつての岩波のグラフ誌『科学の学校』を教科書に，私は兄と2人，毎晩シゴキを受けた。

ほかにも算数や日本の古典がとりあげられたが，かわいそうな兄は，とくに源氏物語54帖の原文すべてを，母の講義とともに1年あまりかけて読破させられた。

母を「教育ママ」のハシリかというと，それもすこしちがうようだ。それに教育が本業だった人に対して，そういう表現は失礼だと思う。家庭内での母のやりかたは，自分は学習しないで子供だけに強要するようなものではなく，場合によっては子供がやらなければそれはかえりみないで，自分だけで勉強を楽しむふうだった。

それだけに強引なやりかたも目立ったけれど，ほかの兄弟たちにはそれなりの成果







illustration：Kyoko Takazawa

があったらしい。たぶん私は「学習ソフト」としての母と相性が悪かったのだ。

父がたった一度だけ，笑いながら私にいった言葉。「お母さんは，集積回路の理論も理解するくせに，じっさいにはヒューズ1本つなごうとしないんだよ」。

父がいおうとしたのは，母が生活上なまけているとか，実践力がないといったことではなくて，学習に熱中するあまり，文字どおりマニアックに走ってしまうことを指していたのだと思う。

もとは母と同業だった父は，若いうちに出版社に「とらばーゆ」して，教育現場の先生がたのための指導書を編集していた。全国の小学校を巡ることも多く，しごとのうえではやはり母の仲間だったのだ。

ただし学校での母は，知識を偏重するような先生ではなかったらしい。音楽も彼女の武器だったので，よくとおる明るい声で音楽のように楽しい授業をしていたと，同僚だったかたから聞いたこともある。

「子供の左ききを，じつに無理なくなおしてもらった」「作文がだいすきになった」「カナヅチだった子供が泳げるようになった」というような話を，いまでも会うたびにする人たちもいる。

いつも学校教育の技術について父と語りあい，職場ではプロとしてそれを実行していた母。その姿勢は家に帰ってわが子に向かうと，無意識に解除されてしまったようだ。すくなくとも子供1人ひとりの特徴に留意することを忘れてしまった。

中西さんふうにツブヤクなら……これはどの職業の親にもあてはまるちょっとしたワナだ。親子でありながら師弟関係を保ちたいなら，相撲部屋の形式を借りる以外にはない……といった感じになるのかな。

## ラドン(Rn)は86番

中西さんの学習ソフトで，何回も行ったりきたりしながら，化学の勉強をした。

ソフトはいつまでも待ってくれるし，わからないと怒りだすということもないから，こわくない。ヒントをたくさんもらっているのに，計算まちがいで「正解」できないときでも，バカにしないで根気よく新しいヒントを出してくれる。

こちらにファイトさえあれば，ソフトを使ってひとりで学習するのは，気が散らなくてなかなかいいと思った。ただし，私の場合はトオルの教科書を片手に，ソフトと併用のかたちでの勉強になった。

物質のすべて，生き物もふくめて，天然のものから人工のものまで，あらゆるものをつくっている基本的な成分が「元素」である。元素は，これ以上，ほかのものに分けることができない。

元素には「原子」と呼ばれる固有の粒子があって，それぞれきまった大きさ，質量，性質がある。水素原子の直径は約1億分の1センチ，ほかの原子も2億～5億分の1センチにしかならない。

原子はさらに細かな粒子で構成されている。中心の「原子核」とそのまわりをいつもとびはねている「電子」である。原子核は原子の大きさにくらべてたいへん小さく，原子の直径の約1万～10万分の1だ。

原子核は正電気を持つ「陽子」と電気を持たない「中性子」からできていて，この2つの数の和が質量数になる。また陽子の数が原子番号となる。

同じ水素原子の中にも，原子核が陽子1個でできている「質量数1」のものと，陽子1個，中性子1個でできている「質量数2」のものがある。99%は前者に入るのだが，このように，原子番号は同じでも質量数のちがう原子同士を「同位体」と呼ぶ。

教科書によると，「同位体は，原子番号が同じであるから，同じ元素に属するものとして扱う。ほとんどの元素には，数種類の同位体が存在する」。

ここまでの初歩の初歩をとりもどすだけでも，たいへんな時間がかかった。それでも，それぞれの元素の「原子の質量」を平均して，「元素の原子量」を算出する実習までこぎつけた。

トオルがときおりやってきて，いろいろ補足をしてくれる。むかし眠りながら母から聞いた話が，じつによくわかる。

原子核のまわりの電子は，いくつかの層になって運動している。これは「電子殻」と呼ばれ内側から順に2，8，18と数がきまっている。数が満たされると外に層をこしらえていくのだが，いちばん外側の電子の運動が，ほかの原子との結合をきめる要因になる。

この「最外殻電子」がもうすこしで満席になるという原子はとても反応性に富み，最外殻にその補数を持っている原子と結合しやすい。まるでブロックあそびだ。

化学反応式も，このブロックあそびを追究したようなものだ。基本を理解して数を合わせ，各項を確認していけば答えが出てくる。

歴史的に名高いアンモニアソーダ製法，「ソルベー法」というパズルのような化学式もトオルにおそわった。アーネスト・ソルベーという人が考案したこの製法は，製造工程でつくられる化合物のすべてが，ふたたび原料に組み込まれるエンドレスの方式なのだ。

中西さんのソフトを横取りしたら，化学が好きになった。学習は与えてもらうよりほしがるほうが効果があるようだ。

「紫式部はとても賢い女性でね，お兄さんが学問の講義を受けているのを，そばで全部覚えてしまったそうだよ」

誰が見たのか知らないけれど，父がそんなことをいっていた。それは紫式部が賢いのではなくて，お兄さんが受けている講義だから，興味を持てたのじゃないのかな。

傍聴と盗聴はだいぶちがうだろうけれど，横取りはやっぱり意欲がわくものだ。

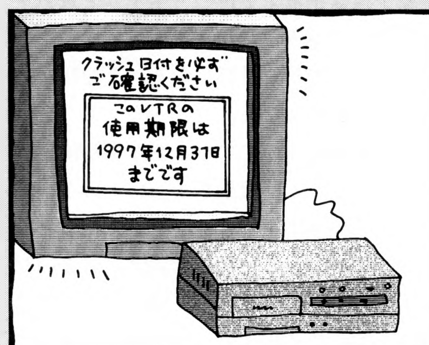
はじめて向学心に燃えている私を，いまや「おばあちゃん」になった母はどう思うかしら。

でも，あのころむりやり母から覚えさせられた元素（当時98種類，現在103種類）の周期律表は，いまでもお経のように身についていて，86番は「ラドン」なんてすぐ出てくるから笑っちゃうなぁ。

## X-OVER・NIGHT
（クロスオーバーナイト）

[第30話]

# 技術革新の体感




TAKAHARA HIDEKI 高原 秀己

ハードディスクが壊れて，かろうじて修理できた話は紹介したが，実はこれに前後して，VTRとCDプレイヤーも壊れてしまったのである！

自宅のモーター関係製品ばかりが次々と壊れてしまうなんて，おそらくみなさんもお聞きになったことはまずないだろう。書いているぼくからして，作り話のような気がしてしかたがない。

モーターのついている家電製品が壊れてしまうような特殊な磁場がぼくの部屋に突然変異的に発生していたのか？　あるいはモーターまわりの制御ICを壊してしまうような特殊電流が流れてしまったのか？などと気持ち悪くなってしまったほどである。

もっとも，VTRもCDプレイヤーも，いずれも「S」のつく社名の製品であり，「S３社の家電製品は壊れやすい」という言い伝えどおりの出来事なので，不思議なことではない。それにこの両方ともすでに数回壊れた実績があり，修理に修理を重ねて使い続けていた。

CDプレイヤーのほうは，オフィスに置いていたポータブルの「ディスクマン」にリプレースするだけにとどめた。もともと，さほど使っていなかったので，別に困ることはない。

問題はVTR。

いろいろと迷った結果，思い切って新しいものに買い替えることにした。修理しても遠からず再発するのは目に見えているし，使っていた機種が何の特殊機能もないシンプルな製品。そろそろ多機能な新製品に切り替える潮時でもあった。

さて，いざ買い替える段になって，とりあえず安い製品でお茶を濁そうと思ったのだが，店頭でいろんな商品の値札を見ているうちに，天の声が聞こえた。

「今回だけはいつもみたいな安物買いの節約はせずに，いろんな機能が揃っているものを買うべきだ……」

世はバブル崩壊による家電製品の価格大暴落時代に突入していたのである。本当にいい時期に買い替えができたといまでも感心しているのだが，S-VHS簡易再生機能つきのBS内蔵ステレオハイファイのVTRが，10万円以内でBSアンテナともども揃えることができた。

いらないと思っていたBSだが，いざ買ってみると，米国の番組をそのまま流していることもあって，英語が洪水のようにテレビから流れてくる。しかも録画したものが２カ国語で再生できるのは，考えていた以上にハイパーな感覚である。もっとも，人によっては５年前から体験していたのだろうが……。

とはいえ，いくら天の声が聞こえたとはいっても，価格が20万円近くしていたのなら，ぼくは今度買ったVTRは買わなかったはずである。改めて家電製品の普及における価格問題は重要である，と実感した。

だが価格問題をある程度度外視しても，いまは次々と現れるハイテク系ニューメディア製品になじむべき時期である，ともぼくは思っている。映像関係ではBSはある程度普及し，次にハイビジョン，CS放送，液晶ビジョン，横長テレビと続々と新しいものが台頭している。オーディオでもステレオAM放送やセント・ギガは無視できず，「録音できるCD」といわれるミニディスク(MD)も登場，対抗馬のデジタルテープ機「DCC」もある。テレコでもデジタルテープ版機が秋から出てきた。

全部を装備するのは資金的にも不可能なのだが，どれかひとつだけでも新しいものを取り入れられれば，少なくとも感覚だけは磨かれる。これはおそらくパソコンをぼくたちが手にしたときに感じた感覚だろうが，定期的に新しいものを使っているという感覚は，ハイテク人種とはいわないまでも，時代の波に乗り遅れないためには，やはり必要なのだろう。

VTRやオーディオに関していえば，適度なサイクルで故障してくれるほうが好都合なのかもしれない。

＊　＊　＊

ところで低価格といえば，三井物産デジタルからのDMがぼくの手元に送られてきた。台湾製のIBM互換機の紹介だった。数日後にこれが新聞の記事になっていた。

「32ビットパソコンが13万円……」

DOS/Vの台頭により，IBM互換機が日本でもようやく障害なしに使えることが認知され，おそまきながら，ようやく本来の「安さ」を前面に打ち出したIBM互換機ビジネスが始まりそうな雰囲気が出てきた。時を同じくして，IBM互換機メーカーのコンパック，デルなどといった米国の大手互換機メーカーも，日本に進出してきた。マスコミも報道合戦を展開しており，ときならぬ「パソコン乱売時代」の雰囲気も出てきたほど。

実際にシステムアップしたときの価格まで含めて考えると，ちょっとどうかとも思うのだが，ローコスト感覚は歓迎すべきことだ。全体が下がれば，これまではゾーンを固定した料金で販売することに何ひとつ問題はない，と考えていたNECやエプソン，シャープ，東芝なども少しは価格を気にした販売に切り替えるだろうから，話はIBM互換機にはとどまらない。

この低価格パソコンブームも，おそらくはバブル崩壊と無関係ではない。これまでは標準的なパソコンの本体価格は30万円前後，という固定観念が確立しすぎており，しかもバブル経済の中で，パソコンは高くても当然だ，という感覚があったことは否めないからだ。しかし全商品にわたる低価格礼賛主義の中で，アンタッチャブルだったパソコンにもこの考え方が浸透し，安いパソコンを世界中から日本に流し込もうという発想が出てきたのだろう。

先ほどの家電の「最先端技術を使おう！」という話と同じことは，もちろんパソコンでもいえる。そろそろぼくも古い「国内標準パソコン」から486マシンあたりに切り替えるべきなのだろう……。

P.S.　明けましておめでとうございます。

料金受取人払

高輪局承認

1396

差出有効期間
平成6年7月
15日まで

郵便はがき

108-00
507

(受取人)

東京都港区高輪
2-19-13 NS高輪ビル

ソフトバンク株式会社

Oh!X 編集部行

□□□-□□ 電話

住所

氏名 年齢

職業・勤務先
学校・学部・学年

●編集部へのメッセージ

今月号の特集について

いちばん良かった記事

興味のなかった記事

これから載せてほしい記事内容

本誌以外にお読みのパソコン雑誌

推薦する市販ソフト

ソフト名：

推薦理由：

付録ディスクに入れてほしいものに何がありますか？

あなたの愛機は(所有機種に○印をつけてください)　　ない
X1(マニアタイプ,C,D,F,G,twin) X1 turbo(model 10,20,30,40,II,III,Z,ZII,ZIII)
MZ-(80K/C, 1200, 700, 1500, 80B, 2000, 2200, 2500, 2861)
X68000(初代,ACE,PRO,PROII,EXPERT,EXPERTII,SUPER,XVI,Compact, HD)
その他　　MIDI楽器(　　)
FD(　基)　TAPE　QD　HD(　MB)　MO　プリンタ(　　)

年齢　歳　パソコン歴　年　男・女　プレゼントNo.

切り取り線

各票の※印欄は、払込人において記載してください。

通常払込料金
加入者負担

# 払込通知票

| | | | |
|---|---|---|---|
| 口座番号 | 東京 1 − 29307 | 金額 | 億 千 百 十 万 千 百 十 円 |
| 加入者名 | ソフトバンク株式会社 | 料金 | 払込み / 特殊 円 |
| 払込人住所氏名 | ※ (郵便番号 ) | 備考 | |
| | | 受付局日付印 | |

この払込通知票は、機械で使用しますので、下部の欄を汚さないよう特に御注意ください。また、本票を折り曲げたりしないでください。 (郵政省)

記載事項を訂正した場合は、その箇所に訂正印を押してください。

切り取らないで郵便局にお出しください。

通常払込料金
加入者負担

# 払込票

| | |
|---|---|
| 口座番号 | 東京 1 − 29307 |
| 加入者名 | ソフトバンク株式会社 |
| 金額 | 億 千 百 十 万 千 百 十 円 |
| 払込人住所氏名 | ※ |
| 備考 | 受付局日付印 |

# 振替用紙

←点線から、きれいに切り取ってご使用ねがいます。

切り取り線

【定期購読のご案内】

●定期購読のお申し込みは、この郵便振替用紙のみとさせて頂き、銀行振込・現金書留によるご入金は、ご遠慮下さい。

●受付締切は、

1日発売　：　発売日前月10日振込
8日発売　：　〃　15日振込
15・18日発売：　〃　25日振込
です。

〈例〉 4月1日発売（Oh！PC 4月15日号）の場合、お振込の締切は3月10日です。

締切に間に合わなかった場合は、自動的に次号からの発送となります。

なお、すでに発売されているもの、また、お振込が締切に間に合わなかった月号のものは、定期購読ではお求めになれません。書店でご購入ください。

●定期購読誌のお届けは書店発売日より遅くなりますのでご了承下さい。

「発売日一覧」

◇毎月1・15日発売
Oh！PC

◇毎月18日発売
Oh！X
Oh！FM TOWNS
C MAGAZINE
Oh！Dyna

◇毎月8日発売
月刊情報処理試験
LANTIMES
THE WINDOWS
DOS／Vmagazine
UNIX USER

●月刊情報処理試験は93年1月号より定期購読料金を改訂させていただきます。お申し込みの際はご注意下さい。

切り取らないで郵便局にお出しください。

この欄は、加入者あての通信にお使いください。

切り取り線

## 送り先

| | | | | |
|---|---|---|---|---|
| お名前 | フリガナ | | お電話 | |
| ご住所 | フリガナ | | | |
| | 〒 | | | |

## 定期購読申込書

| | | | | |
|---|---|---|---|---|
| □Oh！PC | 年間 | (23回) | 12,880円 | (新規／継続 NO.　) |
| □Oh！PC | 6ヶ月 | (12回) | 6,720円 | (新規／継続 NO.　) |
| □Oh！X | 年間 | (12回) | 7,200円 | (新規／継続 NO.　) |
| □Oh！FM TOWNS | 年間 | (12回) | 7,440円 | (新規／継続 NO.　) |
| □C MAGAZINE | 年間 | (12回) | 11,760円 | (新規／継続 NO.　) |
| □Oh！Dyna | 年間 | (12回) | 9,120円 | (新規／継続 NO.　) |
| □月刊情報処理試験 | 年間 | (12回) | 9,360円 | (新規／継続 NO.　) |
| □月刊情報処理試験 | 6ヶ月 | ( 6回) | 4,680円 | (新規／継続 NO.　) |
| □LANTIMES | 年間 | (12回) | 17,760円 | (新規／継続 NO.　) |
| □THE WINDOWS | 年間 | (12回) | 11,760円 | (新規／継続 NO.　) |
| □DOS／Vmagazine | 年間 | (12回) | 9,360円 | (新規／継続 NO.　) |
| □UNIX USER | 年間 | (12回) | 11,760円 | (新規／継続 NO.　) |

〔備考欄〕

この払込通知票は、機械で使用しますので、下部の欄を汚さないよう特に御注意ください。また、本票を折り曲げたりしないでください。　（郵　政　省）

# 愛読者モニタ募集

(M1, M2)

なお,

## M1 コナミ

ピクノ本体価格 29,800円(税別)
モンタージュカード 3,800円(税別)

3名




## M2 シャープ

3名




# 愛読者プレゼント

### プレゼントの応募方法

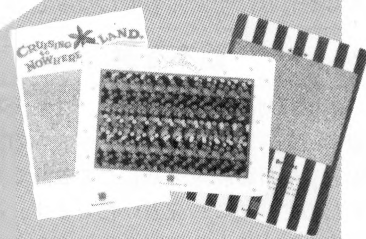
とじ込みのアンケートはがきの該当項目をすべてご記入のうえ，希望するプレゼント番号（P1，P2）をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1993年1月18日の到着分までとします。当選者の発表は1993年3月号で行います。

## P2 3D下敷き

テンヨー
350円(税別)

9名

裸眼立体視が誌面にちらほらと登場する今日この頃。読者の桜井さんからこんなもの売っているよ，というおハガキをいただいたので，さっそく買ってきました。3Dアートですけど，下敷きは平らなので実用性はしっかりあります。

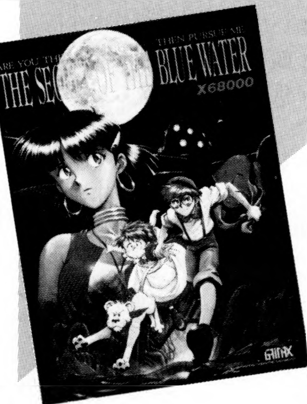


## P1 ふしぎの海のナディア

ガイナックス
☎0422(22)1980

X68000用 5″2HD版

14,800円(税別) 3名




潜水艦ノーチラス号とそれをとりまく人たち。ナディア，ジャン，ネモ船長らが繰り広げる海洋ドラマが「ふしぎの海のナディア」だ。さあ，君もゲーム世界でネオ・アトランティスと戦うのだ。ちなみにこれも要2Mバイト。

### 11月号プレゼント当選者

1ポピュラスII （東京都)藤木健二 （神奈川県)臼井悟士 （福岡県)柴田亨 2ドラゴンクエストV （長野県)宮島雅史 3SX-WINDOWイラスト集Vol.1&2 （新潟県)大津満 （愛知県)角谷光憲 （京都府)加藤真澄 4追補版SX-WINDOWプログラミング （茨城県)外山賢一 （千葉県)川野啓祐 （神奈川県)田中信一 角倉裕之 （福岡県)足達正志 5Sound Windミニ CD （北海道)宝福公司 （福島県)高田英夫 （神奈川県)加藤一八 瀬戸照三 （愛知県)長坂和彦 （京都府)西村武雄 （兵庫県)春名義行 （岡山県)西博之 （広島県)谷川正洋 （大分県)矢上裕之 （敬称略）
以上の方々が当選しました。おめでとうございます。商品は順次発送いたしますが，入荷状況などにより遅れる場合もあります。また，雑誌公正競争規約の定めにより，当選された方はこの号の他の懸賞には当選できない場合がありますのでご了承ください。

# 絵本で知能を作る

## 隠れ研究テーマ

人にはあまり知られていないのですが，実は僕は隠れテーマとして，あることをこそこそとちょっとだけ研究しています。それは絵本です。基本的に被験者1名の幼児に対して絵本を与え，それに対する反応を観察することによって，人の知能の原始的な構成を探り，そして我々が作るべき知能機械に関して示唆となる知見を得ようとしているのです。なんちゃってね。

あることをしたい，目的はこれだ，というところからトップダウン的に知能というものを規定していくというのが，実用的な知能機械を構成する近道のようにも思えます。でも，世の中，そのような美しいアプローチだけではうまくいかないからこそおもしろいのであって，一見，どうってことない幼児期にみせるある反応が，実は知能というものの根源的な意味を表しているということもおおいにありうる話かもしれません。なーんちゃってね。

そこで，今回はその研究から生まれてきた膨大な研究成果のうちの，ほんのさわりだけ(というか絵本の単なる紹介だったりして)を書きます。知能機械を作ろうとしている研究者の方々，あるいは，幼児に絵本をプレゼントしようと考えている方々のどちらかには，たぶん参考になるでしょう。

## 絵本を選ぶ

本屋の幼児向けコーナーには膨大な量の絵本が飾られていますが，ベストの16冊を膨大な解析の結果を基にして独断で選びましたので，ここに発表いたします。絵本専門家のいうことや世間での評判なども参考にしましたし，それよりもなによりも，実際の科学的実験に基づいたデータ(サンプル数1)なので，すごいです。

ベストということばに関して，注意しておくべきことが3つあります。

1)　とりあえず，子供が長期的に喜んだということをよい絵本ということの最大の判断材料にしました。少なくとも子供が欲しているような知的刺激であるのだから，知能というものにもなんらかの好影響があるに違いないという仮定に基づきます。

2)　単に乗り物の写真が並んでいるような絵本類は除外しました。そのような基本的な知識欲というものは，調べるまでもなく学習していく知能機械には必要なものであることがわかっているからです。

3)　対象を1歳から3歳に至るまでの幼児としました。そのころが知能が低レベルから高レベルに向かう時期として大事であるからです。それ以上になると，日本だけとかこの時代だけとかの，環境，社会，道徳的影響がすでに刷り込まれていて，基本的な知識の解明に結びつかないと考えたからです。

## 発表！　絵本ベスト16

### 日本編

**ぐりとぐら**　(福音館書店)

「ぐり」と「ぐら」という2匹のねずみが森の中でお菓子を作り，たくさんの動物たちと食べる話です。古くから名作として知られてきた絵本で，子供の心に深くしみ入るようです。

動物との一体感，料理するという行為，それから主人公のキャラクター，森の中という自然などが混然となって独特の魅力を作っているのでしょう。

**みんなでゆうえんち**　(ポプラ社)

夢のような遊園地の中で遊ぶ話です。両親から離れて迷子になったりしますが，もちろんハッピーエンド。うさぎを誘拐して最後に警官につかまる隠れキャラのねこが実はいたりします。

子供のごく普通の遊び心をとらえます。週末に遊園地に行くよりも，ある意味ではずっとイマジネーションを広げてくれるのかもしれません。

**ぞうくんのさんぽ**　(福音館書店)

動物たちがどんどん象の背中の上に乗っていって，池にみんな落ちてしまうが，水遊びに変わるというのんびりした絵本です。

親しみやすい動物の絵が基本ですが，水に落ちたり，水遊びをするところも重要でしょう。基本的に子宮内における記憶が子供には大人よりも強く残っていて，水に対して安心感を感じるのではないでしょうか。

**ころころころ**　(福音館書店)

色とりどりのボールたちが山道とかジャンプ台とかいろいろなコースをころがっていきます。幾何学的に美しい本です。

重力などの物理法則を確認することに喜びがあるのかもしれません。また，でこぼこ道でボールがぐちゃぐちゃになるところで特に被験者はうれしそうにします。整理整頓するとすっきりするという大人の快感と逆行する快感が実は基本なのかもしれません。

**もこもこもこ**　(文研出版)

地面がなにやらふくらんできて，口のようなところで何か食べたかと思うと今度は丸いものをポトンと落としたりする，抽象的で奇想天外な絵本なのですが，食べる，排泄するというイメージが確かにある絵本です。

パクッと食べるところや排泄物らしきものがポロリと落ちるところで実に被験者はうれしそうな顔をします。食べることと排泄することは，ごくごく基本的な行為ですが，実は対等で重要な欲望なのでしょう。

**おなかのすくさんぽ**　(福音館書店)

動物と一緒に泥まみれになりながら森を探検していきます。最後におなかのすいた動物たちに食われそうになりますが，少年はそうとも知らずに一件落着となります。

本屋に並んでいるほかの美しい絵の本を若いおかあさんなどは買ってしまうのでしょうが，この本の絵は実は人間の中の野性に近い本能を刺激します。まるで，みるみるうちにエネルギーが子供に満たされていくようです。

**ぴょんのたのしいいちにち**　(ブックローン出版)

主人公のぴょんくんが家の屋根を飛びやぶったり，雲で昼寝したり，自由奔放でちょっととぼけた冒険をします。絵はいわゆる「へたうま」に少し近いといえるかもしれませんが，巧妙な作者の計算を感じます。

特にどこにひかれるということはないようですが，とぼけたなかに全体にかもしだされる何かが子供をここちよく刺激します。

**みんなびっくり**　(こぐま社)

象が寝ているあいだに，しっぽのほうに猿がいたずらして顔を描いてしまうという話です。小さい子供にはよくわからないかもしれません。

オーソドックスな題材である動物を主人公にしていますが，視覚パズルを利用することによって，知的な喜びを被験者に与えてくれます。

**ばったくん**　(福音館書店)

これと次の2つは有名な五味太郎氏の作品です。これは，ばったが家の中を飛び回

るというものです。

独特のリズム感，身近な日常風景が子供をひきつけます。色彩豊かな絵も表現力に富んでいます。

**きんぎょがにげた**　（福音館書店）

金魚が家の中のいろいろな風景の中，たとえば，カーテンの模様の中，お菓子の中，水槽の中，おもちゃの中などの似たような模様の中に隠れます。

被験者は目を凝らし，金魚のパターンを一生懸命に探し，みつけると得意げになります。高度な知的快感を刺激されているのでしょう。

**るるるるる**　（偕成社）

飛行機が空をどんどん飛んでいきます。真っ暗な雲の中に入ったり，飛行機の大群の中に入ったりします。

「るるるるる」という言葉遊び，飛行機そのものの魅力も加わっています。急に飛行機の大群の中に入るとアナーキーな快感（先述の反整理整頓快感）を感じるようです。

**外国編**

**あかいふうせん**　（ほるぷ出版）

きれいな線で描かれた幾何学的な絵本です。風船が，花になったり，ちょうちょになったり，傘になったり，連続的に変化します。

子供たちは，ものがいろいろ変わっていくさまをことのほか喜びます。多かれ少なかれ守りに入っている大人ではわからないような，変化の喜びがあるのでしょう。被験者は花が傘に変わるところで声をあげて喜びます，いつも。

**メイシーちゃんベッドにはいります**　（偕成社）

寝るまでの作業，ジュースを飲んだり，歯をみがいたり，トイレにいったりするところを，シミュレートして楽しむしかけ絵本です。

基本的な生活習慣というものを獲得することは，生命を維持するうえで大切なのでしょう。だから，被験者は喜びを感じるのでしょう。

**はらぺこあおむし**　（偕成社）

青虫が子供の喜ぶおいしい食べ物を食べていくうちについにきれいなちょうちょに変わるという話です。きわめてきれいな絵とともに，くだものを食べるシーンの本の作り方には感心させられます。

この本は，子供の食欲の中枢，それから，ものの変化を喜ぶという中枢をあざやかに刺激して，とりこにします。

**とべ，かえる，とべ**　（評論社）

かえるがほかの天敵の動物たちから間一髪のジャンプで逃げていくという話です。最後，人間の子供につかまってしまいますが……。

次から次へとおそいかかる動物たちから逃れようとするかえるに感情移入します。怖いという気持ちが大きければ大きいほど，逃げられてよかったという喜びは増すようです。

**まよなかにきたライオン**　（ミキハウス）

ライオンがお城の中の赤ちゃんを連れ出し，どんどん逃げていきます。てんやわんやの大騒動も，城の中でおいしいものをたくさん食べて一件落着します。

ライオンがにこにこしている赤ちゃんを背中に乗せてぐんぐん進み，それを王様をはじめとする一団が追いかけるという状況設定自体が子供の心をひきつけるようです。でもやはり，ものを食べまくるシーンをいちばん喜びます。

## 絵本の分析

今回紹介してきた絵本たちのどういうところが幼児の知能を刺激し快感を与えてきたのかと考えまして，とりあえず，それぞれの絵本の持つ要素をピックアップしました。それが，以下の10個です。

- 恐怖（怖いものから逃げて助かる喜び）
- 運動（動きのダイナミズムは生み出す喜び）
- 動物（動物との一体感から生まれる喜び）
- 自然（自然の中ですごす喜び）
- 日常生活（通常の生活をシミュレートする喜び）
- 物理法則（ものの運動を通して物理法則を体感する喜び）
- 食/排泄（食べたり排泄する喜び）
- パズル（視覚的パズルを解く喜び）
- 変化（ものが変化するのを見る喜び）
- 水（水自体への喜び，子宮の思い出？）

各作品ごとに，そのテーマが強く出ているものを◎，ある程度以上出ているものを○で，表にまとめました。日本とアメリカでそれぞれ有名な作品「ぐりとぐら」「はらぺこあおむし」は多くのテーマを含んでいることがわかります。それだけ，総合的に訴えてくるのでしょう。「おなかのすくさんぽ」も盛りだくさんですね。

あともうひとつ，ここで紹介した16作品のどれかを買ってみようという人のために，フローチャートを載せておきます。まず，子供に知的な刺激を与えたいか，動的な刺激を与えたいかという枝分かれから始めてください。4つほど質問に答えて枝分かれすると，この16作品のどれかにたどりつきます。それがおすすめということです。

**ベスト16作品フローチャート**

Start
知的な刺激
動的な刺激
動物との連帯感
Yes
No
ワクワクする
ほんわか楽しむ
パズル的な楽しさ
日常生活の楽しさ
ストーリーの楽しさ→まよなかにきたライオン
エネルギーが湧き出る→おなかのすくさんぽ
ズッと余韻が残る→ぐりとぐら
なじみやすい→ぞうくんのさんぽ
ストーリーの楽しさ→みんなびっくり
感覚的に楽しむ→きんぎょがにげた
平日の喜び→メイシーちゃん……
週末の喜び→みんなでゆうえんち
はねまわる喜び
Yes
No
明るい楽しさ
逃げる喜び
物体の変化の喜び
物体の運動の喜び
ストーリーの楽しさ→ぴょんのたのしいいちにち
家の中を探検する→ばったくん
逃げる喜び→とべ，かえる，とべ
幾何学的な楽しさ→あかいふうせん
排泄の喜び→もこもこもこ
食べる楽しさ→はらぺこあおむし
幾何学的喜び→ころころころ
飛行機の喜び→るるるるる

**ベスト16作品の要素分析**

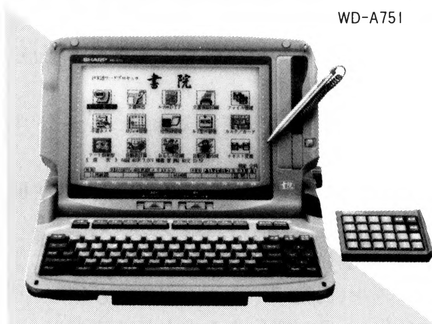
| タイトル | 恐怖 | 運動 | 動物 | 自然 | 日常生活 | 物理法則 | 食/排泄 | パズル | 変化 | 水 |
|---|---|---|---|---|---|---|---|---|---|---|
| みんなでゆうえんち | | ○ | ○ | | ◎ | | | | | |
| ぐりとぐら | | | ◎ | ◎ | ○ | | ○ | | ○ | |
| メイシーちゃん…… | | | | | ◎ | | ○ | ○ | ○ | |
| ぞうくんのさんぽ | | ○ | ◎ | | | | | | | ○ |
| はらぺこあおむし | | | ○ | ○ | | | ○ | ○ | ◎ | |
| もこもこもこ | | | | | | | ◎ | ○ | ◎ | |
| おなかのすくさんぽ | ○ | ◎ | ○ | ○ | | | ○ | | | ○ |
| みんなびっくり | | | ○ | | | | | ○ | | |
| ばったくん | | ◎ | | | ○ | ○ | | | | |
| きんぎょがにげた | | | | | ○ | | | ◎ | | |
| あかいふうせん | | | | ○ | ○ | | ○ | ◎ | ◎ | |
| ころころころ | | ◎ | | | | ◎ | | ○ | | |
| るるるるる | | ○ | | | | ○ | | ○ | | |
| とべ，かえる，とべ | ◎ | ○ | ○ | ○ | | | | | | ○ |
| ぴょんのたのしいいちにち | | ◎ | | ○ | ○ | | | ○ | | |
| まよなかにきたライオン | ○ | ◎ | ○ | | | | ○ | | | |

# PENGUIN INFORMATION CORNER

## ペ・ン・ギ・ン・情・報・コ・ー・ナ・ー

**NEW PRODUCTS**

パーソナルワープロ
### WD-A751/761
シャープ


WD-A751

シャープは，専用のペンを使い「手書き編集（ジェスチャー）」を搭載したパーソナルワープロ「WD-A751」「WD-A761」を発売した。
「手書き編集」機能は，文字消去，スペース，アンダーライン，移動，複写などの基本編集機能を14種類の記号を使い，ペンオペレーションで実現するもの。そのほかにも「手書き文字入力」「手書きメモ」などの各種アプリケーションで，ペンオペレーションによるシステムが採用されている。

搭載アプリケーションには，イラストや飾り罫を自由に組み合わせてオリジナルカードや文書を作成できる「アート倶楽部-ペン」，手紙，表計算，グラフ，罫線，レイアウトをこなせる5つの「らくらくソフト」，ポップ印刷，テープラベル印刷，のし紙印刷など10パターンの決まったフォーマットを簡単に印刷できる「おもしろ印刷Ver.5」がある。

また，「WD-A751」は明朝体，毛筆体，ゴシック体のスーパーアウトラインフォントを内蔵。400DPIの高品位印字に加え，熱転写ながら毎秒100文字の印字ができる。「WD-A761」ではさらに丸ゴシック体を加えた4書体を内蔵し，印字速度は「WD-A751」に比べ1割ほど高速化されている。また，類語辞書の搭載，2400bpsの小型カードモデム対応の通信ソフトを標準装備（カードモデムは別売り）している。

価格は「WD-A751」が235,000円，「WD-A761」が265,000円（ともに税別）となっている。
〈問い合わせ先〉
シャープ㈱　☎03(3260)1161,06(621)1221

タッチパネル操作のBTRONパソコン
### 電房具1B/desktop
パーソナルメディア


1B/desktop

パーソナルメディアでは，BTRONパソコンの新シリーズとして「1B/desktop」を発売した。

本機は，BTRON仕様に準拠したOS「1B」をデスクトップパソコンに実装したものである。そして，従来の操作系を継承しつつ，今回の「1B/desktop」では，アプリケーションのユーザーインタフェイスに，電子ペンやマウスだけでなく，タッチパネルによる操作がアプリケーションに依存せず可能となった。メインCPUに80386SX(20MHz)を使用し，メインメモリは標準で5Mバイト(最大13Mバイト)，640×400ドット(16色)のグラフィック，3.5インチドライブを2機搭載している。

本体価格は，40Mバイトハードディスク内蔵タイプ「1B/desktop-40」が448,000円，100Mバイトハードディスク内蔵タイプ「1B/desktop-100」が498,000円，タッチパネル付き14インチディスプレイ「1B-TP14」が350,000円（すべて税別）。

また，1B/desktopの発売を記念し，最大24回払いに対して金利負担をサービスする「1B/desktopキャンペーン」を12月末まで行っている。
〈問い合わせ先〉
パーソナルメディア㈱　☎03(5702)0355

シリコンディスク
### SDP-2.5/10/20,SDI-10/20/40
エプソン販売



SDI-5

エプソン販売では，米国サンディスク社と販売代理店契約を結び，サンディスク社製「シリコンディスク」を発売した。

今回発売するモデルは，1.8インチIDEドライブタイプ「SDIシリーズ」と，軽量，携帯性を重視したJEIDA Ver.4.1準拠カードタイプ「SDPシリーズ」の2タイプ。

本シリーズは，512バイト単位のブロックでデータを消去できるフラッシュメモリチップを使用し，ハードディスクと同様の管理ができるとともに，ECC(エラー訂正)とチップごとの書き換え回数の長寿命化を実現した。

半導体メモリを使用した記憶媒体であるため，軽量，対衝撃性，耐振動，低電圧駆動などの特徴を備えている。

価格は，「SDP-2.5(2.6Mバイト)」が65,000円，「SDP-10(10Mバイト)」が145,000円，「SDP-20(20Mバイト)」が210,000円，「SDI-10(10.4Mバイト)」が150,000円，「SDI-20(20.9Mバイト)」が215,000円，「SDI-40(41.9Mバイト)」が420,000円となっている。
〈問い合わせ先〉
エプソン販売㈱　☎03(3377)2465

## ハンディプロジェクタ
# HP-A1
富士写真フイルム

HP-A1




富士写真フイルムでは，ハンディプロジェクタ「HP-A1」を発売した。

本機は，１型，約89,000画素の液晶パネルを使った液晶プロジェクタであり，最大70インチまでの大画面映写を行える。

また，ステレオスピーカーを内蔵。ビデオ信号で接続するため，ビデオ出力端子のあるAV機器を簡単に接続して，映像を楽しむことができる。

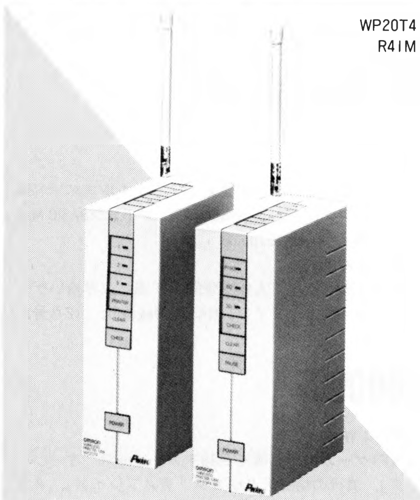
外形寸法は，222mm(幅)×83mm(高さ)×224mm(奥行)で，本体重量は2,300gとなっている。

価格は98,000円（税別）。

〈問い合わせ先〉
富士写真フイルム㈱　☎03(3406)2981

## ワイヤレスプリンタリンク
# WP20R2-256/R4-1M/T2/T4
オムロン

WP20T4
R4IM




オムロンは，パソコンからプリンタへ印字データを無線伝送できる，ワイヤレスプリンタリンク「WP20シリーズ」を発売した。無線方式に周波数400MHz帯の特定小電力無線を使い，見通し距離で50mのデータ転送ができる。データ転送速度は4800bpsであり，子機から親機の転送に独自のデータ圧縮を行うことによって効率的なデータ転送が可能である。

そして，プリンタにつなぐ親機，パソコンにつなぐ子機ともに標準タイプで２個，拡張タイプなら４個のコネクタを搭載。親機についているコネクタのひとつは，プリンタ接続用でほかのコネクタはパソコン接続用であり，親機，子機ともに拡張タイプを使えば，最大プリンタ１台に対して７台のパソコンを接続できる。

また，親機には標準タイプで256Kバイト，拡張タイプで1Mバイトのデータバッファをもっている。

価格は，親機の「WP20R2-256(標準)」が69,000円，「WP20R4-1M(拡張)」が79,000円，子機の「WP20T2(標準)」が59,000円，「WP20T4(拡張)」が64,800円（すべて税別）である。

〈問い合わせ先〉
オムロン㈱　☎03(5488)3221,06(282)2672

## ファイル転送ユーティリティ
# XIN/XOUT II ver.7.0e
電机本舗

XIN/XOUT II ver.7.0e




電机本舗では，MacintoshとDOSマシンをRS-232Cでつなぎファイル転送を行う「XIN/XOUTII」のバージョンアップを行った。

今回のバージョンアップでは，1Mバイト以上のファイル転送時に，System7のディスクキャッシュ機能と衝突しないように改善している。これによりディスクベースでは転送不可能な巨大な画像ファイルなどの送受信が可能である。

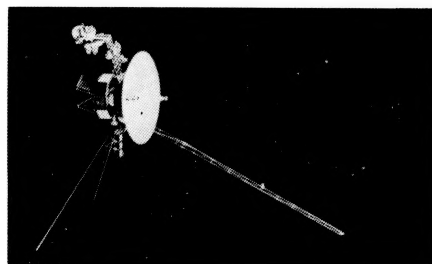
また，新たに支援ユーティリティとして，簡易エディタ「Bullet」，データ圧縮「Squeeze」，マックバイナリ展開「MACBIN」を標準装備することになった。

価格は，従来どおり14,800円（税別）。

〈問い合わせ先〉
㈲電机本舗　☎03(3447)1773

## INFORMATION

## TEPIA第5回展示
# HUMAN IN SPACE
TEPIA



TEPIA（財団法人機械産業記念事業団）では，平成４年10月９日～平成５年３月12日まで，TEPIA第５回展示「HUMAN IN SPACE」を港区青山の産業記念会館で行う。

現在，人類が初めて宇宙飛行してからすでに４半世紀がたち，それにともなって宇宙利用技術も飛躍的に発展した。本展示は，こうした成果をふまえ宇宙を利用する技術を多くの人々に認識してもらうことを目的に行われる。展示内容は，

1)　SPACE VEHICLE
トランスポーテーション・ビークルの飛行技術の紹介
2)　ON THE EARTH
地球上での宇宙開発技術の紹介，リモートセンシング技術によるデータ解析された映像をインタラクティブ操作で体験できる
3)　IN SPACE
宇宙利用技術としてマイクログラビティ(微小重力環境)でのさまざまな実験や具体例の紹介
4)　SATELLITE
人工衛星に関する技術，軌道衛星の紹介
5)　SPACE RESOURCES&NEW ENERGY
宇宙における新エネルギーの開発技術
6)　YAC SHILD
日本宇宙少年団の活動紹介，パソコンによる宇宙シミュレーション学習コーナー
7)　LIFE IN SPACE
宇宙生活を送るための将来構想の紹介

以上，７つのゾーンと本展示の全体像を映像で紹介するイントロダクションシアター，宇宙開発の軌跡と宇宙利用の将来像，そして，ドキュメンタリーやドラマ映像を上映するスペースシアターの２つのミニシアターもある。

開館時間は，平日が10:00～18:00，土曜・日曜・祝祭日が10:00～17:00で，毎週月曜日は休館。入場は無料である。

〈問い合わせ先〉
TEPIA第５回展示事務局　☎03(3226)8356

# FILES Oh!X

このインデックスは，タイトル，注記――著者名，誌名，月号，ページで構成されています。あけましておめでとう。1993年，今年の目標はもう立てましたか？　去年まではできなかった，新しい何かができるようになるといいね。

**参考文献**
I/O　工学社
ASCII　アスキー
コンプティーク　角川書店
C Magazine　ソフトバンク
テクノポリス　徳間書店
POPCOM　小学館
マイコンBASIC Magazine　電波新聞社
My Computer Magazine　電波新聞社
LOGIN　アスキー

## 一般

▶アルゴリズムを見切ったぞ!?
画面の効果の巻その2。効果的な画面切り替えのサンプルプログラムを紹介。――おにおん，テクノポリス，12月号，120-124pp.

▶THE NEWS FILE
コンパックの低価格パソコン日本上陸，データショウ'92，堀井雄二氏へのインタビューなど，パソコン業界周辺の話題。――編集部，LOGIN，21号，34-41pp.

▶3次元コンピューターグラフィックへの招待！
ヨンセンマンのCFなど，3次元CGの最先端の映像を紹介。CGアーティスト原田大三郎氏らへのインタビュー。3次元CGの作り方の手順を解説。――編集部，LOGIN，21号，219-235pp.

▶ワープロ/パソコン通信新聞
NIFTY-Serve「チケット予約販売」など最新ネット情報満載。短期連載・パソコン通信への道。草の根BBS情報。――山本まさこ，マイコンBASIC Magazine，12月号，94-98pp.

▶BASICプログラミング講座
「ハノイの塔」のプログラミングで再帰の原理と理論を学ぶ。――東　幸太，マイコンBASIC Magazine，12月号，104-108pp.

▶V.R.EXPO'92開催
10月に名古屋国際会議場で行われた日本初のバーチャルリアリティの展示会。日本と海外の取り組みの模様をレポート。――福富忠和，ASCII，12月号，200p.

▶社会の木鐸
10月分の雑誌・新聞の記事から，ハイテク関連情報を抜粋して紹介。DOS/Vパソコンの日本上陸による低価格戦争への動きが目立つ。――鬼怒川厳＆甘粕英，ASCII，12月号，218-219pp.

▶Digi-Ana Valley
オーディオのデジタル技術とアナログ技術の接点の研究の連載。今月は，貼ると音のよくなるシートなど，CDの音質向上への取り組みを紹介。――編集部，ASCII，12月号，245-252pp.

▶年末年始パソコン活用術
パソコンで年賀状やカードを作るためのガイド。石田博也氏のX68000による年賀状作成法も登場。――編集部，ASCII，12月号，269-292pp.

▶ことば遊び・コンピュータ
精神科医の真似をするソフトELIZAから，コンピュータによる日本語の会話プログラムの可能性を考える。――ホーテンス・S・エンドウ，ASCII，12月号，357-360pp.

▶バカパパのモノを買い物
レーダー作戦ゲームのスタンダード版から効果音鳴り響く豪華版まで。ほかに「残日計」など。――バカパパ，ASCII，12月号，386-387pp.

▶ラッキー！ハッピー！オッケー！
論文や小説は承諾なしで引用できるが，音楽著作物は許諾をとって使用料を払わなければならない。これは正当か？　弁護士に訊くシリーズ18回目。――編集部，ASCII，12月号，408p.

▶ワープロの身軽さを生かしてつくるクリスマスカード＆年賀状
年賀状やカード作成に利用できるワープロの機能などを紹介。シャープ書院ほか。――編集部，My Computer Magazine，12月号，71-77pp.

▶マイコンからMy Computerへ
創刊15周年記念企画。ダイナウェアの藤井展之氏など業界のキーマンへのインタビューを通して今後のパソコン界の動きを探る。――編集部，My Computer Magazine，12月号，78-91pp.

▶エレクトロニクスショー'92による最新技術トレンド
10月のインテックス大阪でのエレクトロニクスショウ'92のレポート。CD-I，MD&DCC，ペンコンピュータなどの出展紹介を通じ，今後の業界の動向を伝える。――高橋雄一，My Computer Magazine，12月号，94-103pp.

▶PC実験室
さまざまな環境でのマウスの耐性を検証。コーヒー，消しゴムのかすなど，たいていの汚れはこまめな掃除でとれるとか。――石川至知，My Computer Magazine，12月号，144-148pp.

▶ビジネスマンのための情報管理室
シャープの電子手帳の活用ノウハウとして，HAL-CATCH Ver.2の概要を解説。――塚田洋一，My Computer Magazine，12月号，174-177pp.

▶MYCOM WATCHING
農地の賃借関係のほか，貸借を希望する農家同士の仲介を務めるシステム，FICSによって農家の経営規模の拡大を図ろうとする試みをレポートする。――菊地秀一，My Computer Magazine，12月号，230-233pp.

▶横浜発・コンピュータ化された外資系ホテル
インテリジェントホテル「横浜グランドインターコンチネンタル」を紹介。カードキーを使用し，ミニバーやレストランの使用額などをフロントで集計・解析。――大窪志保，My Computer Magazine，12月号，238-241pp.

▶なんでもQ＆A
シャープのワープロ「書院」とDOS/Vパソコンを一体化した「書院パソコン」。パソコンモードで画面の階調を変更する方法，ワープロモードでMS-DOSファイルを呼び出した場合の利用方法などについて回答。――シャープ，My Computer Magazine，12月号，286-287pp.

▶パソコン言語A to Z
言語ソフトを解明する特集。言語ソフトの目的やインタプリタ，コンパイラの解説に始まり，Pascalの概要，C++のテクニック伝授などを満載。――青山一朗ほか，I/O，12月号，17-42pp.

▶スーパーコンピューティング入門
数学パズルシリーズ第4回。4色問題と呼ばれる命題について，これまでの証明への取り組みを紹介，コンピュータの果たした役割も考える。――林智雄，I/O，12月号，167-170pp.

▶書院パソコン
書院パソコンの仕様と，そのメリットや使い方を解説。――編集部，I/O，12月号，92-93pp.

## MZシリーズ

**MZ-700/1500 (S-BASIC)**

▶ALICE IN MAZELAND
ハートを矢印方向へ誘導しゴールへ。全20面のアクションパズル。――村田達也，マイコンBASIC Magazine，12月号，127-128pp.

**MZ-2500(BASIC-M25)**

▶LAST one
盤上の白いボールを，ある法則で取り除いていく。名作パズルの移植版。――工藤俊介，マイコンBASIC Magazine，12月号，129-131pp.

## X1/turbo/Z

**X1シリーズ**

▶STAR DUST
降ってくる星から地球を救え！　ミサイルコマンド風シューティングゲーム。――KENTARO，マイコンBASIC Magazine，12月号，156-157pp.

▶田吾作さん
どろぼうモグラから人参を奪回。穴掘り人参拾いゲーム。――楠　則夫，マイコンBASIC Magazine，12月号，158-159pp.

## X68000

▶GAMING WORLD
国産F1ゲームの最高峰を目指すカーレース「オーバーテイク」，流行の格闘アクション「デスブレイド」，8方向スクロールアクション「バーンウェルト」，ほかに「エトワール・プリンセス」，「サンダーレスキュー」を紹介。――編集部，テクノポリス，12月号，20-25pp.

▶燃える攻略野郎
「ポピュラスII」の攻略。能力の振り分け，面の序盤から敵の攻撃まで。攻略法を解説。――編集部，テクノポリス，12月号，66-69pp.

▶SOFT EXPRESS

「オーバーテイク」，「デスブレイド」，「エトワール・プリンセス」。──編集部，コンプティーク，12月号，67-75pp.

▶特集/光栄

光栄のシミュレーションゲームを紹介。「三國志III」の攻略法や期待高まる新作情報。──編集部，コンプティーク，12月号，94-99pp.

▶Software Hot Press

新着ゲーム紹介。宇宙海賊の魔の手から地球を救え「サンダーレスキュー」，アクロバットアクションで敵を倒せ！「ストライダー飛竜」，キュートでコミカルなアクション「エトワール・プリンセス」，超リアルF1カーレース「オーバーテイク」，神秘の異次元で繰り広げられるアクション「バーンウェルト」。──編集部，POPCOM，12月号，24-27pp.

▶こだわりレポート

「ポピュラスII」。神の気分を味わっちゃう不思議で美しいシミュレーションをレビュー。──不破淳郎，POPCOM，12月号，76-77pp.

▶NEW SOFT

グローディアのX68000オリジナル「バーンウェルト」。──編集部，LOGIN，21号，26p.

▶最新ゲーム徹底解剖!!

ズームの期待作「オーバーテイク」のすべてのサーキットコースとテクニックを紹介。シミュレーションゲーム「ネクタリス」の後半部の解説。──編集部，LOGIN，21号，130-159pp.

▶X68000新聞

新着ゲーム「ストライダー飛竜」，「沈黙の艦隊」，「ロードス島戦記II」の紹介。C言語講座。──編集部，LOGIN，21号，244-247pp.

▶ZETTON

魔法使いの見習いゼットンは，伝説の妖精ファリスを探し求めて……。大きなキャラクタが魅力のスクロールアクション。──林　純一，マイコンBASIC Magazine，12月号，161-162pp.

▶マリ夫の冒険　PART2

マリ夫を落とさないように操作してカギを集める。2つのキャラを別々に動かすアクションゲーム。──高橋秀之，マイコンBASIC Magazine，12月号，163-165pp.

▶Out Run ～Magical Sound Shower～

セガのゲームミュージックプログラム。要NAGDRV(CM-64，SN-U110-10)。──山賀　求，マイコンBASIC Magazine，12月号，176-177pp.

▶誌上公開質問状

ディスプレイ「CZ-604D」でTVを見るには，X68000マウスの修理は，などの質問に回答。──多田太郎，マイコンBASIC Magazine，12月号，178p.

▶PRODUCTS SHOWCASE

マウス操作可能の低価格レイトレーシングソフト「MIRAGE System Model Stuff」をレビュー。手ごろな価格で3Dグラフィックスを体験できる。──編集部，ASCII，12月号，293-304pp.

▶FREE SOFTWARE INDEX

主要パソコンネットにアップロードされたソフトから選んで紹介。X68000用はTF.X，IPD.Xなど。──編集部，ASCII，12月号，421-427pp.

▶なんでもQ＆A

キヤノンBJ-10VはX68000で利用できるか，SX-WINDOW用のプログラム開発ツールの内容は，などの質問にシャープが答える。──シャープAVCシステム事業推進室，My Computer Magazine，12月号，284-285pp.

▶HOBBY EXPRESS

イマジニアの「ポピュラスII」と日コン連企画の「キャノンサイト」をゲームレビュー。そのほか「シミュレーションゲームにおける鉄道の役割」など。──相川春利，松田浩二ほか，My Computer Magazine，12月号，323-339pp.

▶98でフォーマットされた5インチ光磁気ディスクをHumanでアクセス

PC-9801のMS-DOSフォーマットの5インチ光磁気ディスクをHuman68kで利用するデバイスドライバ。──市原昌文，I/O，12月号，62-64pp.

▶GAME BOX

カプコンの「ストライダー飛竜」のレビュー。──伊藤ゆうほか，I/O，12月号，82-87pp.

▶Merry Final

誰でも気軽に楽しめるアクションパズルゲーム。画面にいるネズミをブロックで囲い込め！──土方嘉徳，I/O，12月号，95-98pp.

▶GCCで学ぶX68ゲームプログラミング

先月号の付録ディスクに収録のX68000版G＋＋を受けて，G＋＋でスプライトを扱う。今回はクラスの定義方法とコンストラクタについて。──吉野智興，C Magazine，12月号，150-155pp.

▶Information from Compiler Makers

シャープからのインフォメーション。SX-WINDOW開発キットのなかのサンプルプログラムの内容と目的について。──シャープXグループ，C Magazine，12月号，166p.

# ポケコン

**PC-E500**

▶ROAD

ジグザグ道路をはみ出さないように動く。繊細ドット操作ゲーム。　──M-99，マイコンBASIC Magazine，12月号，167p.

## 新刊書案内




**コンピュータに夫や妻を奪われた人のための公式支援ハンドブック**
**コンピュータ・ウイドー/ウイドワーを支援する専門家集団著**
**小林　達/栗本　武訳**
**HBJ出版局刊**
**☎03(3234)3911**
**B5変形判　231ページ**
**1,400円(税込)**

日本語のパソコンおたく，英語でいうとコンピュータナード。パソコンマニアがおたく呼ばわりされて異人視されるのは日本に限ったことではないのだ。むしろ，アメリカのほうが深刻かもしれない。なにしろあちらは，奥さんの相手をせずに仕事や趣味に没頭したりしたら離婚されかねない国なのだ。だからコンピュータウイドーなどという言葉もできてしまう。コンピュータウイドーっていうのは，旦那がパソコンの相手ばかりしていてこれじゃいないと同じだわ，っていう奥さんのこと。確かに，パソコンマニアと結婚した奥さんは相当の覚悟が必要だ，というのは想像に難くない。そのあたりは先月紹介した『プログラマの妻たち』に詳しい。不幸なのは結婚したあとで，旦那がパソコンマニアになってしまうこと。日本では「奥さんが許可してくれないから新しいパソコンを買えない」って泣き言をよく聞くが，許可しない奥さんは賢明である。が，子供を作りたくない場合には許可したほうがいい。完璧な避妊法になるからだ(いかん，本書のノリがうつっている)。

さてさて，本書はCWWA，コンピュータ・ウイドー/ウィドワーズを支援する専門家集団によるコンピュータ中毒者を配偶者に持った人の神経を逆撫でする本である。コンピュータウイドーに対し，「あなたは絶対にコンピュータには勝てないから，あきらめてコンピュータと仲良くなるよう努力しなさい」ってささやくからだ。せいぜい，家庭をコンピュータに完全に支配されないためにはコンセントのない部屋を作りなさい，って程度。確かにそれは正しいけれども。

はてさて，もちろん，CWWAなどという団体はない。これはかのうっとうしいアメリカンジョークの本なのだ。「アルゴリズム硬化症」や「プログラマ脳症」，などのくだりは傑作だが，半分くらいは笑い転げられるほどではない。まあ，アメリカンジョークだから暇つぶしとしてはよいぞ。　(K)

---

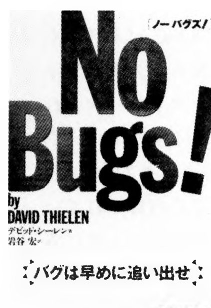



**IMONを創る**
**いがらしみきお著**
**アスキー出版局刊**
**☎03(3486)1977**
**A5判　215ページ**
**1,500円(税込)**

IMONとはナンだろう？　誰でもまず，最初にそう思うだろう。ナンといっても著者はあのシマリスくんやスナドリネコさんの登場する「ぼのぼの」の作者である。スーパーマンごっこしているみたいなIMOの絵の表紙もちょっとアヤシイし……。

この「IMON」はTRONに対抗した「壮大なOSプロジェクト」だという。そして，この『IMONを創る』では，IMON計画に基づいたさまざまなコンピュータ論が展開する。とはいえ，そこはいがらし氏，決して「論」といった堅苦しさはない。

そう，I(いつでも)M(もっと)O(おもしろく)N(ないとなぁ)なのである。




**No Bugs!**
**デビッド・シーレン著**
**岩谷　宏訳**
**ビレッジセンター出版局刊**
**☎0424(88)9421**
**A5判　241ページ**
**3,200円(税込)**

著者は，MicrosoftでMS-DOS Ver.5.0などの開発を担当したデベロッパである。そして，本書はずばりタイトルどおり「バグとり」のための方法論である。しかし，単なる試用段階でのデバッグのやり方ではなく「バグの早期発見＆早期治療対策集」なのである。

著者の提案は，プログラマがコーディングをしている段階で極力バグフィックスしてしまうべきだという，いわば「同時進行型デバッグ」。

主としてC言語によるプログラミングが前提として述べられているが，ほかの言語で開発を行っている人にもおおいに参考になるに違いない。

# QUESTION and ANSWER



**私は半年前にX68000を購入したばかりです。いろいろとわからないことが多いのでどうか教えてください。**

**Oh!Xの1992年6月号の143ページからの「DBTL.X」ですが，プログラムの入力方法がわかりません。誰にでもわかるようにやさしく教えてください。それと要アセンブラ，リンカとありますが，どのようなものなのでしょうか。電脳倶楽部のハイスピードアセンブラ，ハイスピードリンカのことでしょうか。**

**Oh!Xの1992年11月号の32ページからの「SAVESC.SYS」ですが，これも同じくプログラムの入力方法を教えてください。**

**X68000XVIにハードディスクを取り付けたいのですが，LHD-FM100E(ロジテック)，EFX-100B(エニックス)，HD-J100(システムサコム)のどれでも使用できるのでしょうか。　　　　山口県　松井**



アセンブラは，Cコンパイラに付属のAS.Xまたは電脳倶楽部に収録されたHAS.X，リンカはCコンパイラ付属のLK.Xまたは電脳倶楽部に収録されたHLK.Xが使えます。HAS.X，HLK.XはAS.X，LK.Xに比べて機能拡張と処理速度の高速化が図られていますのでおすすめです。

しかし，実際のアセンブル作業には開発ツールのほかにIOCSCALL.MAC，DOSCALL.MACというファイルが必要です。IOCSCALL.MAC，DOSCALL.MACはCコンパイラをお持ちならシステムディスク1のINCLUDEディレクトリに含まれています。Oh!Xの過去の付録ディスクや『X68000マシン語プログラミンググラフィック編』の付属ディスクなどに収録されています。以上のファイルがひとつでも欠けるとDBTL.Xを作成することができません。

それではX68000XVIに付属のシステムディスク（SX-WINDOWではないほう）とブランクディスク（または内容が破壊されてもいいディスク）を1枚用意してください。システムディスクをドライブ0に，ブランクディスクをドライブ1に入れて，OPT.1キーを押しながらX68000の電源を入れてください。ドライブ0のシステムディスクからHuman68kが起動しましたか？ドライブ0がAドライブ，ドライブ1がBドライブに割り当てられていますね。ドライブ1のブランクディスクをフォーマットしますので，

　FORMAT B:

を実行してください。フォーマットが終了すると，

　別のディスクをフォーマットしますか？

と聞いてきますから，“N”を押してください。

質問から離れた話になりますが，システムディスクのバックアップを取っていないようなら，ここでバックアップを取っておきましょう。もう1枚ブランクディスクを用意して，いま説明した手順でフォーマットします。マスターのシステムディスクをドライブ0に，フォーマットしたブランクディスクをドライブ1に入れて，

　DISKCOPY A: B:

を実行してください。ドライブ1のディスクにシステムディスクの全内容がコピーされます。以後システムの起動はバックアップしたディスクで行い，マスターディスクは大切に保管しておきましょう。

話を質問の回答に戻します。ドライブ1にフォーマットしたブランクディスクを入れてありますね？　次にIOCSCALL.MAC，DOSCALL.MAC，アセンブラ(AS.XまたはHAS.X)，リンカ(LK.XまたはHLK.X)をドライブ1のディスクにコピーします。コピーにはCOPYコマンドを使います。COPYコマンドの使い方がわかからないようなら，Human68kユーザーズマニュアルをご覧ください。

リストの入力には「エディタ」という編集ソフトを使います。エディタはシステムディスクのBINディレクトリにED.Xのファイル名で収められていますので，

　ED B:DBTL.S

を実行してください。

では1992年6月号144ページのリスト1を入力しましょう。エディタの使い方はHuman68kユーザーズマニュアルに書かれています。

リスト1は説明の便宜上行番号を印字してありますが，行番号を入力する必要はありません。たとえば13行は，

　_KEY_INIT equu　　$03

だけ入力します。ところどころ空白が入っていますが，スペースキーでなくTABキーを使って入力します。スペースキーを何回も叩くより入力の手間が省けますし，ファイルサイズも小さくなります。13行の入力の手順を箇条書きにすると，

　_KEY_INIT
　TABを押す
　equ
　TABを押す
　$03
　リターンを押す

となります。

リストを最後まで入力，または途中で中断したいときは，ESCを押してからEを押すと入力したリストをセーブしてエディタを終了します。次回起動したときに，

　ED B:DBTL.S

を実行すると前回の続きから入力を再開できます。また途中まで入力したリストをセーブしたあとも，入力を続けるときはESCを押してからHを押します。

リストをすべて入力し終わったらカレントドライブをBドライブにして，

　B:
　AS /W DBTL

を実行してください（ハイスピードアセンブラを使うなら，ASの部分をHASに置き換えてください）。リストが正しく入力できていれば，

　No fatal error(s)

と表示されます。エラーが表示された場合はエラー行をエディタで訂正します。

ひとつのエラーもなくアセンブル作業が終了したらリンクします。本来リンクはいくつかのオブジェクトファイルをひとつにする作業です。DBTL.Sのように単体のプログラムにリンク作業は不要のように思えますが，アセンブラで書いたプログラムは，アセンブル→リンクの手順を踏む決まりになっています。リンクを行うには，

　B>HLK DBTL

を実行してください。エラーがなければ，BドライブにDBTL.Xが作成されます。

　B>DBTL

でディスクバトラーを起動します。万が一起動しなかったり，おかしな動作があったらリストの打ち間違いがあると思われます。リストが正しく入力されているか確認してください。アセンブル，リンクでエラーが表示されないのに動作がおかしい場合は，文法的には合っていてもリストを1行飛ばしているとか，d4レジスタをd1レジスタと読み間違って入力したりといったことが

考えられます（常駐物があればそれとの相性が悪いとも考えられますが）。入力したプログラムと掲載されたリストを見比べて，誤って入力した箇所を探します。骨の折れる作業ですが頑張ってください。

次に本誌11月号のSAVESC.SYSの入力方法について説明します。Oh!X92年6月号をお持ちのようですから，付録ディスクに収録されているMAC.Xを使ってリスト4を入力することができます。付録ディスクの解凍の手順は1992年6月号の41ページに詳しく書かれています。目的のMAC.Xは解凍して作成されるディスク1のPROGディレクトリに収録されています。AS.XなどをコピーしたディスクにMAC.Xをコピーします。

リスト4はLHA.Xで圧縮されたかたちで掲載されていますので，ファイルを解凍するにはLHA.Xが必要です。1992年6月号の付録ディスクにLHA.Xも収録されていますので，MAC.Xと一緒にコピーしておいてください。

MAC.XはCAPSキーをロックしていると，エディットモードで正しく動作しません。CAPSキーのランプが赤く点灯していたら，CAPSキーをもう一度押してランプを消しておいてください。

MAC.Xを置いたディスクをBドライブに入れて，

```
B:
MAC
```

を実行してください。

```
New file(y or n)
```

と聞いてきます。新規ファイルを作成しますので“Y”を押します。次にファイル名を聞いてきますので，これにはSAVESC.LZHと入力します。雑誌に掲載されているのと同じようなダンプリストが画面に表示されたら，“C”（CRC ON）と“E”（エディットモード）を押します。リストの先頭の4桁の数字は入力する必要はありません。たとえばリスト4の最初は，

```
20 D0 2D 6C 68 31 2D 97
```

の部分だけ入力します。1行入力したら画面右に表示されている値（チェックサムといいます）と，リスト4の E6 を比較します。同じ値でなければ，いま入力した部分に誤りがありますので，見直して訂正します。チェックサムは縦にもあります。1画面分入力したら縦サム，縦サムと横サムの交点にあるCRCが掲載されたものと同じか確認してください。

すべてを入力し終わったか，途中で中断するときはESCを押してコマンドモードにしてからSを押します。ファイルサイズ2850バイトを指定してセーブしてください。ファイルを解凍するには，

```
B:
LHA E SAVESC.LZH
```

を実行してください。

もしファイルが解凍できない場合は，入力間違いがあると思われますので，再度MAC.Xを起動してチェックサムを確認します。今度は新規作成ではないので，

```
New file(y or n)
```

には必ず“N”を押してください（中断した入力を後で再開するときも“N”を押します)。“Y”を押すと入力したファイルが確実に破壊されます。気をつけてください。

最後にハードディスクについての質問がありますが，確実なのはX68000を購入した販売店に「X68000に××のHDはつながりますか？」と尋ねることです。不安があるならメーカーからX68000対応として発売されているものを選ぶのが無難でしょう。

対応品でない場合，特定の機種が接続できるかどうかについては実際に接続してみる以外に手はありません。理論上はいくつかの目安があるのですが，確実に動作保証することはできません。



**アセンブラで書いたたくさんのファイルをアセンブルしてリンクしようとすると，**

Relative addressing overflow in ……

**と表示されてリンクができません。どうしたらいいんでしょう？　東京都　大林　隆**



アセンブラマニュアル第5章5節「LKエラーメッセージ一覧」を見ますと，相対アドレスの相対値が最大値を超えた場合に表示されるエラーのようだとわかります。bsrやbra命令は飛び先を実行中のPCからの相対値で表します。相対値は符号付き16ビットで示しますので，−32768〜32767の範囲を表すことができます。外部参照を用いてbsr，braの分岐元と分岐先が2つのファイルに分割されていて，飛び先がこの範囲を超えたときに質問にあるエラーが発生します。ちなみに外部参照を用いずに，ひとつのファイル内で分岐先が相対値で表すことのできる範囲を超えた場合は，アセンブルエラーが発生します。

対応策としてはbsrをjsr，braをjmp，相対で届きそうもない相対アドレッシング命令を絶対アドレッシング命令に置き換えるか，リンクする順番を変えてみます。リンクする順番でエラーが発生しなければ，実行に差し支えはありません。

68000のアセンブラは相対アドレッシングを使えるものが多いので，大規模なプログラムを複数のファイルに分けて作成するケースで，リンク時の相対エラーを目にすることがあります。bsr braがエラーの原因なら変更も簡単ですが，たとえば，

```
lea.l work(pc),a1
```

というような命令があって，workがPCからの相対範囲で表せない場所にあったりすると，

```
lea.l work,a1
```

のように書き換える作業が必要です。

相対アドレッシングは絶対アドレッシングに比べてオブジェクトコードが小さく，実行サイクルが速いので，好んで使う方がいるかもしれませんが，符号付き16ビット幅で収まる範囲にあるのかどうか少しだけでも注意してプログラムを組むと，あとで変なエラーが出て面倒な思いをせずにすみます。

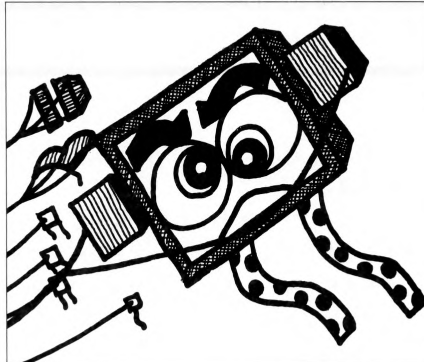
（影山　裕昭）

**質問にお答えします**

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を挙げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに解答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので電話番号も明記してくださいね。

**宛先：〒108　東京都港区高輪2-19-13**
**NS高輪ビル**
**ソフトバンク株式会社出版部**
**Oh!X編集部「Oh!X質問箱」係**

STUDIO X ON AIR



# FROM READERS TO THE EDITOR

**1993年の始まりです。どんなことが起こるかな。どんなことをしようかな。何かを始めるのもいいし，いままでやってきたことを，初心に戻って見直すのもいいですね。今年は酉年。みなさんにとってよい年でありますように。**

◆先日の大雨で雨漏りが起こり，Mさんよりも大切な愛機XVIに直撃！　……水もしたたるいいパソコンになってしまった。修理から帰ってきたXVIにはビニールコートをかけ，ついでに地震対策の命綱もつけることにした。これで，もう何も怖くないぞ！(嘘)

でも，いまどき雨漏りする部屋に下宿しているボクって……。　北川　亮(22)東京都

レインコートを着ているX68000って珍しいなあ。ところで，Mさんってだあれ？

◆11月号はいつもと違い，京都出身だ。見学旅行中に買ってしまった。渡辺　洋平(16)北海道

ちゃんと「見学」のほうもしましたか？

◆私の大学は，コンピュータ設備が自慢である。しかしながら，いまあちこちでウイルスが発見されて，システムの「総点検中」になってしまった。11月号のX-OVER•NIGHTの話は，その意味で実に共感した。デュアルシステムはコストがかかるが，システムの保守能力は，(オンラインでない限り)実用性は高い。データを各自フロッピー上に持っているので，被害のほどはわからないが，早く復旧してほしいものである。でないと宿題がぁ……。菊池　重幸(19)千葉県

◆「転ばぬ先の杖」という言葉を聞いてディスクのバックアップをとる，間違ってマスターを消す。初心者のときこんなことやりませんでしたか？　私はやった。狩野　太郎(19)神奈川県

◆ハードディスクのバックアップをとっておいたフロッピーが昇天した。それとは知らずシステム再構築のためハードディスクをフォーマットしてしまい……。毎晩，涙で枕を濡らしております。　大内　良介(17)東京都

「事故」はいろいろなところで起こるものなのですね。そこで，今月の格言は「天災は忘れた頃にやってくる」です。データ管理には細心の注意を払いましょう。

◆受験生のみなさん！　封印の期限はすでに過ぎています。まだ封印していない人はすみやかにX68000を押入れにしまい，家族のいる方は封印宣言をしておきましょう。仲間がいると思えば封印生活も少しは楽になるハズです(涙)。

小山内　将剛(20)青森県

受験もそろそろ追い込み。がんばって乗り切れば，プログラミングにゲームに，ますます楽しいX68000ライフが待ってますよ。

◆本屋に積んであったOh!Xを上から3冊目ぐらいのをとって買った。家に帰ってみると，なんか異物感が。おや，別冊付録がついてたのか。えーとなになに。

「京都周辺みどころMAP完全保存版」

なんか違うよーな気がするがOh!Xのことだしぃ～，と思ったが，よく見ると「る○ぶ11月号別冊付録」。なるほど，だから阪神は優勝できなかったのか。　金子　孝康(19)茨城県

なるほど，だからりえちゃんは貴花田関と結婚するのか。

◆ゲームはこれからもどんどんリアルな方向に進んでいくだろう。しかし，究極にリアルなゲーム，プレイヤーの五感をすべて制御したバーチャルリアリティなんかは，現実に私たちが生きているこの世界とどこが違うのだろうか。ここに現実の世界があるのに，もうひとつ現実の世界を作り上げて，何の意味があるのか。究極のバーチャルリアリティを私たちはいま体験しているのに，別の世界へ逃げようとしているのか。いずれ，バーチャルリアリティのなかにまたバーチャルリアリティが現れることだろう。すでに私たちの世界も何番目かのバーチャルリアリティかもしれない。　吉本　康孝(20)福岡県

本当の自分はどこにいるのでしょうね。

◆神社や公園などにはたくさん鳩がいるのに，鳩の死骸を見たことがない。もしかして，象の墓場ならぬ鳩の墓場が人知れぬところにあるのではないでしょうか。　経澤　重徳(22)東京都

鳩の墓場だったら，象の墓場ほど大きくなさそうだから，狭い日本にも結構たくさんあるかも……？

◆'87　初代X68000を見て，大学に入ったらバイトして買うことを決意。

'88～'89　モンモンと毎日を送る。

'90春　1浪したらX68000も5年たち，32ビット機が出るのではと思い，浪人を決意。

'91春　去年の計算間違いに気がつき，自分のバカさかげんにあきれて2浪へ突入。

'92春　期待した5年目の回答が小さくなっただけだったため失意のまま3浪へ。

というわけで，シャープさんへお願い。来春の新製品がラップトップだったりしたら4浪しちゃいそうなので，32ビット機を出して！

野島　明憲(21)愛知県

おやおや，でも新製品への要望は人さまざま。次のハガキのような人も……。

◆体がブチ壊れて1か月以上も入院している。X68000NOTEでもあれば持ち込めるのに。

松口　淳(22)埼玉県

◆前略　シャープ様。

ノートX68000を早く出してください。そしてそれを私の大学の建設学科海洋工学に売り込んでください。さもないと某○EC社のPC-9801NS/Tを買わされてしまいます。年内中にご配慮を賜りたいと存じます。　草々

宮野　文武(19)神奈川県

これこれ，年内中なんて無理いっても……。いずれにしても，わくわくするような新機種が出てくるのが楽しみですね。

◆フロッピーが増えすぎてMOが欲しくなった今日この頃ですが，3.5インチモデルにするか，




◀橋本　和典　東京都
橋本さんの描く女の子って，「キュート」って感じ！　12月号に間に合わなかったサンタ嬢のカードは担当者の机上に飾ってあるのです……えへへ。




◀玉野　健一　奈良県
こちらの彼女は「チャーミング」って言葉がぴったりかな。繊細だけど元気な感じがイイですねぇ。どっちが好みかって？　そりゃあ，どっちも！

５インチモデルにするかで迷っています。3.5インチモデルのほうがアクセス速度や値段，今後の主流になる，などで有利かと思うのですが，５インチモデルの大容量も捨て難いし……。

ところで11月号では，巻末のアクセスの広告が先月号の小人さんから魔人さんに変わっていましたが，小人さんは計算を間違えてしまったのでしょうか。　長石　裕行(23)長野県

小人さんはクビになったのでしょうか。いまは不景気だし，この寒空に路頭に迷っていたらかわいそう……。

◆７月，貯蓄が臨界量を超えたために発生したブラックホールは，その後も相当量の電磁波を放射しつつ，月末になるとこれまた大量の金銭を吸い込んで成長を続けている。先日はディスクが吸い込まれたまま出てこなくなってちょっと焦った(ラベルのはがれ)。そして私は，困ったことに前期試験中での脱出に失敗してしまった。こういう生活ができるのも今年１年をバイトの神様に捧げたおかげだと思う。

ところで本物のブラックホールって，マンガみたいに成長したりしないんでしたよね。
要　秀紀(20)京都府

「貯蓄が臨界量を超えた」って，もしかして，とってもとってもお金持ちになれた，ってことかしらん？　たとえ一瞬でもそうなれたらいいなあ。

◆子育てに役立つプログラムがあったら教えてください。　渡邊　久理子(20)富山県

どんなことに使えるでしょうねぇ？　どれだけ食べさせたら，どのくらい太るかのシミュレーションとか(ちょっと違うか……)。でも，真面目な話，そういう分野ってこれからどんどん開拓されていくような気もしますね。誰か実用ソフトを作りませんか。いっとくけど，某「子育てシミュレーションゲーム」のことじゃありませんよ。

◆友人から有償で譲り受けたもう１台のX68000を学級の子供たち(小学生)に開放しています。いまはZ'sSTAFFでお絵描きして遊んでいます。「マリオペイントみたい」といわれ，カチンときたりしましたが。　田口　瑞穂(30)秋田県

そのうち，マリオペイントとかのほうが，X68000のソフトみたい，っていわれるようになったら，なんとなくうれしいなぁ。

◆DIS.Xを利用したオプティマイザを作った。しかし，最適化によってかせぐ時間よりも最適化自体にかかる時間のほうが大きい。あんまり意味がないかな。まあ，サイズも少し小さくなるし，アマなんだからいいか，とプログラムをいろいろ改造しています。尹　忠秀(23)長崎県

いまは効率が悪くても，そういう日々の努力の積み重ねで，いつかすごいものを作れるようになるかもしれません。

◆僕の隣の部屋の友達に，東京のある楽器店からYAMAHAのTG100が届いた。箱についていた伝票をふと見てみると「取鳥県取鳥市」になっていた。人口がいちばん少ないとはいえ……。やっぱり「山の陰」だからだろうか？




◀清水　健太郎　愛知県
清水さんは今度，大学受験とのこと。合格してどんどんイラスト描いて投稿してくださいね。いや別に浪人してても投稿は，歓迎ですけれど，ねっ。




◀姉帯　寛　茨城県
澄み切った空気が伝わってくるようですね。空の色はぎらぎらした真青じゃなくて，透明感のあるきれいな水色で「すうっと高い」って感じかな。

P.S.　みなさん，間違えないように。
井戸　滋(18)鳥取県

ひ，ひどい。砂丘だってあるいいとこなのに(関係ないって)。でもちゃんと届いてよかったですね。ワープロの漢字変換だったら絶対間違えないのにね。

◆先日，バーチャルリアリティなるものを体験してきました。鉄砲(？)を撃って相手をバラバラにするやつで，結構快感でした。でも歩くのがボタン操作なのには参りまして，結果，乗り物酔いの激しい私はこれで酔ってしまいました。今度やるときは酔い止めを持っていったほうがよいかも!?　高野　将成(16)山口県

これから遊びにいく人，身に覚え(？)があったらご注意！　まだ，乗り物酔いしたことない私は安心，安心……だけど，遊ぶヒマが……くすん。

◆都民から信州上田の住人になった。給料は減ったが，生活環境は大幅に向上した。ここからゴミゴミした東京を見下ろすのは痛快の極みであったりする。一極集中万歳。
藤原　利治(25)長野県

ゴミゴミした東京の，さらにゴミゴミした編集部で働く私たち……。なに？　ちゃんと掃除しなさいって？　う～ん，返す言葉もございません。

◆GT-6000を買った。Z'sSTAFF PRO-68Kも買った。GT-6000用のSCSIボードも買った。明日から，生きてゆけるだろうか？
……Z'sSTAFFを開封して，目が点になった。3.5インチのディスクが出てきたからだ。ツァイトに送れば，５インチに交換してくれるだろうか？　岡部　和秀(23)愛知県

とうとう，X68000の世界にもそういう時代がやってきたんですね。両方のドライブを持っている人はまだ少数派のようですし，みんな，ソフトを買うときには注意してね。

◆私の先輩にセガの社員がいます。彼は「バーチャレーシング」の開発にいたんですが，先日そのゲームをやりに行ったとき，彼は「画面モードが違う！」といって，店員に直させてしまった。さすが，作った人はこだわりが違うなぁ(彼はハトやウシを作ったそーです)。

木島　智(22)宮城県

えらい！　開発者の鑑！　とお伝えください。自分の仕事にこだわりを持つって，大事ですよね。そういう人が作ったものなら安心できる気がします。

◆祝！　X68000「カルトＱ」出演！　11月１日(日)のカルトＱにX68000が出ていたとのことで(私はマヌーなことに風呂に入っておった)，そのとき家族がいっせいに画面を指さして「おとーさんのパソコン！」と叫んだそーだ。しかし，家族の誰も問題は覚えていなかった。何じゃ，こいつらは……(ところでどんな問題だったんですか？)。　柳井　敏彦(34)愛媛県

「おとーさんのパソコン」って，なんかほのぼのしてていいなあ。で，問題のほうですが，ちゃんと「報告」がきております。

◆11月１日の夜，カルトＱ(コンピュータゲーム特集)を見ていると，「さてここからは超カルト問題で得点が２倍になります。第１問，ポピュラスIIが日本に最初に移植された機種は？」。……このとき，６年間読んできたOh!MZ，Oh!Xが普通の雑誌ではないことをあらためて知ったのだった。　八谷　忠男(19)広島県

う～ん，カルトはともかく，「超」カルトとは……。

◆結婚してからパソコンは買えないと思い，寸前にXVIを買ったが，僕が彼女以外のことに夢中になるとすぐにプリプリしだして，なかなかプログラムする時間がなくて困っています。シューティング以外で彼女とすると楽しいゲームって何がいいですかね？　２人同時プレイゲーム特集をしてください。　仁井内　明(23)広島県

「彼女と遊ぶゲーム特集」かあ。いいかもしれない。ちょっと某ポ○イ誌みたいだけど。ところで，↓のような人も……。

◆このあいだ，彼女にＨなゲームが見つかってしまった。それ以来，彼女は僕の……僕のXVIに会いにくる。さらに先日，○○パート２なるソフトを買ってきて「火星にかわってせっかんよ！」とか，どこかで聞いたようなことをいいながらムチをふるっていた。こんな２人っていったい……。　森山　健史(21)福岡県

し・あ・わ・せ……なんでしょ？

◆先日，昔に写した８ミリを見ていたら，同一人物の現在と４カ月前が写っていた。母が４カ月前の人物を見て「この人，感じいいね」と言った。そのあと現在のその人を見て「この人，意地悪そうね」と言った。母は同一人物ということに気づかなかった。ちょっとの間にこんなに人が変わることってありますか？

川上　良太(17)埼玉県

さて，その４カ月の間に彼or彼女にナニが起こったのか？

◆パソコンを買ってすぐの頃，用語がまったくわからなくて，紙に片っ端から書いて壁に貼っていました(いまも貼ってある)。説明を見つけたら書き込んで，またわからない用語が出てきたら書き込んで，かなりの量になってます。パラメータ，ルーチン，プロンプト，カスタマイズ……と，とにかく言葉がずら一っと並んでいるのを見ると，記憶力に欠ける私としては非常に不安になってしまいます。もうじき歴３年になるのに……ほとんど理解しないまま何となくパソコンいじくってる。ううう。

岩瀬　貴代美(20)福岡県

「う～ん，えらい！」と思っちゃいました。編集部あてに，「初心者なので，どうしたらいいかわからない」というハガキをたくさんいただくのですが，そういう方たち，参考になったでしょうか。やっぱり，地道な努力がいちばん大切かもね。

◆「ファイト！」「オー」
「ファイト！」「オー」
誰か，寝言で声をかけあう場面にあったことのある人がいるだろうか。私はある。気持ちが悪いが，大爆笑ものである。

藤原　彰人(22)岡山県

寝る間も惜しんでトレーニング？

◆最近，学力と模試の結果との間にかなり強力なカオスが発生している。入試までに「カオスの法則」が見つかればなんとかなる……わけない。

中安　規雄(19)奈良県

これこれ，「カオスの法則」なんか探してないで，勉強するのぢゃぞ。

◆あのぅ，どうでもいいようなことなんですけど，Ver.xxxって何を基準に決めるんです？ なんだか気になって。

間宮　義晴(17)山形県

大きな仕様変更や機能追加があるときは１増えて，そうでもないときは小数点以下の増加になる，ってのが一般的見解のようです。でも，「Ver.3.38659」とか出てきたら，やっぱりやだなあ……。

◆１日が30時間くらいあればよいと思う。

東尾　治紀(23)大阪府

そーですよねぇ。でも，そのときに人間の活動スピードが0.8倍(つまり遅くなるわけね)になったとしたら，いまと同じですよね。ということは，いまの1.25倍のスピードで動けば，１日は30時間になったようなもの！　よし！　……と思ったのですが，そんなに高速で生活したら疲れちゃって休息時間もたくさん必要かもしれないし……。あああ，どうすればいいの？

◆いまの幼稚園児はゴレンジャーは知らないと思う。

和田　正晴(20)千葉県

◆そういえば「ピンポンパン」を知らない世代というものが存在しているんですよね。この話を友達にしたら，「最後の木の中のオモチャを取っていくのがうらやましかった」といっていた(同感)。ちなみに僕の実家にはピンポンパンの最終回のビデオがあるんですが，β(ベータ)なのでいまは見られないんです(このために買おうとまでは思わない)。はっ，もしや，ベータを知らない人もいるのでは？　くわばらくわばら。

黒畑　喜弘(19)新潟県

ちょっと前に「いまの中学生(たぶんいまの高校生の人たちだろーなあ)は，ピンクレディー(古い！)を知らないんだって」という話題がショーゲキ的に出回ったことがあるのですが，その歌ばかりか振り付けまで覚えてる私って……。でもでもゴレンジャーもポワトリンもセーラームーンも知ってる……。意味はないけど，ふと「少年老い易く学成り難し」なんて思っちゃったりして。くわばらくわばら。

◆大学の研究室にパソコンが１台持ち込まれた。機種はというと，PC-9801F(FAでもFSでもFXでもなくF)という(超)アンティークなマシン。みんなで「最新のFシリーズだぜぃ」といって嘘ついてまわっている。

小川　靖浩(21)東京都

それはですね，ゲームなどに熱中せずに研究に専念できるように，という温かいご配慮があるのですよ，きっと。でも，まさかメディアは８インチのみ，なんてことはないですよねぇ……？　だったら完璧！

◆いま，立体視が流行ってるようですが，片目でTVを見ると立体に見えるって知ってましたか？

真田　百人(22)北海道

どっちの目でもいいんですよね？

◆てきとうにやっているようで，ちゃんとそれらしく見える立体視。これだけでも，なんとなくMATIERが欲しくなってくる。荻窪氏が考えることもすごいが，ちゃんと形にしてくれるMATIERもすごい。

小海　崇史(22)千葉県

◆いままで何度も「ステレオ写真」を見かけましたが，ほとんど立体に見えたことがありませんでした。が，11月号の「大人のためのX68000」の「目いっぱい」はわりと楽に立体視できました。ぼくはかなり強い近視(たぶん0.01以下)なので，眼鏡をはずして，目から15cmぐらいのところにOh!Xを置いてやると簡単です。

林　秀明(22)宮城県

◆「大人のためのX68000」はMATIERを使用して手軽にできるので，なかなか面白かった。どうでもいいが，「見つめる公園」の写真の背景，木々が黒くて何だかわからん。私はてっきり月面の写真かと思ったぞ。月面に飛来する目型UFO。ああ，恐ろしや……。

八木沢　良二(18)栃木県

◆「目いっぱい」は非常にシュールでよかった。

岡田　和久(24)京都府

◆おおっ！　立体に見える！　というわけで生まれて初めて立体視というのに成功しました。これも荻窪さんのおかげです。

五十嵐　正治(18)東京都

◆裸眼立体視をやっているところを見られたら，さぞまぬけな顔をしているだろうなぁ。

倉田　泰幸(22)茨城県

そーいえば，「デート中の禁止事項その３：裸眼立体視をしないこと」って，デートのマニュアル本に載って……(嘘だよん)。

◆ここ数年，漫画を読むことが多かったが，最近は小説(特に池波正太郎)を読んでいます。そのかわりにOh!Xを読まなくなったけど。編集部のみなさん，これは正しいことですよね？

円福　貴光(18)福岡県

ぴくっ(眉間にシワが寄った音)。「Oh!Xを読まなくなった」が「正しい」……？
ぴくぴくぴく……。わりゃあだれにものゆうとんじゃい！　なんてはいいませんけどやっぱりOh!Xも読んでほしい……なぁ。

◆いま，空から降ってきてほしいもの
1)　ハードディスク
2)　単位
3)　体育の日の雨
もし実現したら，天気予報に釘付けになるだろうな。

矢元　章夫(19)兵庫県

いちおー社会人してるワタクシといたしま




▲溝畑　知幸　兵庫県
バックアップ取っていてよかったですね。CDの下敷でディスクが壊れたなんて……。編集部も，掃除したら昇天したディスクが発見されるかも。




▲藤沢　実　東京都
「みんな仲よし」うんうん，そーでなくっちゃ。これからももっともっと，みんなで一緒ににぎやかにX68000の世界を広げていきましょうね！

しては，とりあえず2)はどーでもよいのですが，1)だったら，走って取りにいくでしょうねぇ。やっぱり。

◆X68000のクロックUP改造をするのは，好きな女の子に告白するのと同じくらい勇気がいる(まだどちらもやってないけど)。

天達　雄一(17)京都府

X68000は改造しなくても動くけど，女の子はちゃんと告白しなくちゃ誰かに取られちゃうかも!?　がんばって！

◆11月号52ページのはみだしに載っていた坊農誠くん。便器にフタをすれば済むんでは？　本でも板でも使って。

それはそうと，MZ-700，X1turbo，X68000と名機を所持してるうえにパソ通までやってる身分で貧乏ネタ出しても誰も同情しちゃくれんぞ。

山田　俊英(24)東京都

あとのほうは誰のことかわかんないけど，同感！　それはそうと便器のフタにOh!Xは使わないでね。お願いぃぃぃ。

◆アセンブラでプログラムを作っているとき，データを書き込むのが最高に気持ちいい。これは，イラストを描いているとき，最後にハイライトを入れる瞬間の快感に似ている。

伴　武士(21)千葉県

◆プログラミングもまた，ひとつのゲージツですね！　河合　竜次(18)岐阜県

動作やプログラムそのものの美しさを目指し，それが完成したときのヨロコビ！

◆今年も郵便局の年賀状配達の募集の季節がやってきました。MIDIを買うためにがんばりたいと思います。　氷見　孝(17)富山県

◆さー！　今年の冬も高輪郵便局でアルバイトだ！　木村　亮(19)静岡県

この号が発売される頃は，もう区分け作業をしているのかな？　編集部あての年賀状もたくさんあるといいなぁ。……みなさん，お待ちしておりますよっ。

◆鍋がおいしい季節になりましたね。

高橋　学(19)京都府

「鍋に熱燗」は冬の醍醐味！　だねっ。

◆先日，初雪が降りました。生まれたときから雪国にいるとはいえ，初雪とは特別な感じのするものです。これが毎日のように降りだすと，うっとうしくなるのですが，たくさん積もらないとスキーができないというジレンマになるわけです。　米田　孝(22)北海道

雪かきや雪おろしはたいへんだけど，雪景色は美しいし，スキーもできるし，雪国の人，ちょっとうらやましいなあ。積もった朝には，真っ白でまだ誰も歩いてないところに足跡をつける快感，もあるしね。



◀占部　哲彦　広島県
彼女は何をお願いしているのでしょうか？　こんな可愛い顔で「ねっ，お願い」なんてささやかれたら，「ダメ」なんて言えない……よね。

## ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。
●ソフトの売買，交換については，いっさい掲載できません。
●取り引きについては当編集部では責任を負いかねます。
●応募者多数の場合，掲載できないこともあります。
●紹介を希望されるサークルは必ず会誌の見本を送ってください。

### 仲間

★「THE FINAL MAX-SPEED」では，ディスクマガジンの強化をするためにスタッフおよび，新規会員を募集します。ディスクマガジンには，MEW氏制作のオリジナルシェルやMOPMDRV，スタッフの平木敬太郎氏によるオリジナルフォントを使用するなど，ディスクマガジンとしてのベースは固まっています。そして，オリジナルレーベルもあり会員の自作ソフトを発行しています。また，独自のネット開局も予定しています。ミュージック，CG，プログラミングのできる方，原稿の書ける方，編集スタッフ，読者など幅広く募集します。興味のある方は，300円分の為替か現金を下記の住所にお送りください。折り返し，紹介用に制作したディスクマガジンをお送りします。〒610-01　京都府城陽市寺田樋尻69-60　office THE FINAL MAX-SPEED

### 売ります

★X1用FM音源ボード「CZ-8BS1」，X1用データレコーダ「CZ-8RL1」をそれぞれ13,000円前後で売ります。付属品はありますが箱はありません。まずは，往復ハガキに希望価格を書いてお送りください。〒350-13　埼玉県狭山市狭間山台2-24-104　木下　卓也(20)

★シャープ製プリンタ「CZ-8PG2」を70,000円で売ります。箱，付属品すべてあります。連絡は往復ハガキでお願いします。〒243　神奈川県厚木市妻田北1-14-14-1423　山野　和也(22)

★Roland製MIDI音源モジュール「CM-64」＋ギターカード＋X68000用MIDIボード「SX-68MII」を70,000円前後で売ります。箱，付属品あり，2カ月使用の新品同様です。連絡は往復ハガキでお願いします。〒389-08　長野県更級郡上山田町温泉2-25-7　山崎　高志

★X68000用MIDIボード「SX-68MII」＋Roland製MIDI音源モジュール「SC-55」を60,000円で売ります。箱，マニュアル，付属品すべてあり。新品同様です。連絡は往復ハガキでお願いします。〒312　茨城県勝田市東石川2-24-1　松本　実(17)

★X68000用内蔵1Mバイト増設RAMボード「PIO-6BE1-A」を10,000円で売ります。また，X1用FM音源ボード「CZ-8BS1」を5,000円で売ります。なお，「PIO-6BE1-A」は未使用の新品です。連絡は往復ハガキでお願いします。〒243　神奈川県厚木市吾妻町1-18-301　戸塚　康弘(21)

★Roland製MIDI音源モジュール「CM-32L」を送料込み，33,000円で売ります。完動品で箱以外はすべてあります。連絡は往復ハガキでお願いします。〒369-03　埼玉県児玉郡上里町三町542　岡村　哲男(18)

### 買います

★シャープ24ピン漢字プリンタ「CZ-8PK7」または「CZ-8PK9」「CZ-8PK10」を送料込み20,000〜30,000円で買います。プリンタケーブルはなしでも可。連絡は往復ハガキでお願いします。〒737　広島県呉市弥生町6-33　谷本　和生(39)

★「試験に出るX1」を送料込み5,000円で買います。なるべく美品を希望。連絡は官製ハガキでお願いします。〒803　福岡県北九州市小倉北区今町3-18-21　成　孝徳(19)

### バックナンバー

★Oh!X1989年2月号を3,000円で買います。折れ，多少の汚れはかまいませんが，切り抜きがあるものは不可です。連絡は往復ハガキでお願いします。〒457　愛知県名古屋市南区中割町4-89県営中割住宅404号　神野　力(17)

★Oh!X1988年9月号と「試験に出るX1」を各1,000円で買います。連絡は往復ハガキでお願いします。〒514　三重県津市上浜町2-102　ABマンション3-C　田村　晃一郎(28)

# 編集室から from E・D・I・T・O・R

## DRIVE ON

このコーナーでは，本誌年間モニタの方々のご意見を紹介しています。今月は11月号の内容に関するレポートです。

●スプライトの使い方，キャラクタの管理など，考えるだけで頭が痛くなりそうな部分についてよくわかる特集でした。パソコンを買ってから3年以上もたっているのに，いまだゲームプログラムが完成しない自分にとっては，雑誌にゲームプログラムを投稿できる人がすごくうらやましいです。今度は“どうすれば作りかけのプログラムが完成するのか”について特集してくれるとありがたいですね。
塩谷　望(19)　MSX2　茨城県

●11月号の特集でいちばん印象に残ったのは，「180300クロックの死闘」ですね。以前，スプライトでゲームを作ったときは，ここまで追求しませんでしたが，走査線との戦いはやってましたからね。唸りながら読みましたよ。アセンブラプログラミングじゃあ日常茶飯事ですから最近はこういった戦いも快感です。「ああ，ここももっと速くなるやんか」とかいいながらプログラミングするのは，高級言語じゃあ味わえません。でも，アマチュアだからこんないい方をするんでしょうね。プロの方は「1バイト1クロックに命を削る」なんでしょうね。せっかく，青本を持っていることだし，私自身も精進していきましょう。
中矢　史朗(21)　X68000 ACE-HD　愛媛県

●11月号の特集を読んで，ゲームプログラムにとって信頼性は二の次，なのにちゃんと動かなくてはならないというものだと思いました。よく考えてみれば不思議なことですね。よりスマートに，より確実にといったところでしょうか。そのために洗練されたテクニックが必要なのはもちろん，根性とか思いつきもいるんでしょうね。少ないメモリでいかに動かすか，遅いCPUでいかに処理を高速に行うか。「これでもか！」という根性がないと，たぶんゲーム屋にはなれないでしょう。私は，なんとなく「ゲーム屋にはなりたくない」と思っていました。できることなら恵まれた環境でプログラミングをしたいからです。
安井　百合江(18)　X68000 PRO　愛知県

●11月号のX-OVER・NIGHT「バックアップ」は，私も何度か悔しい思いをしたことがあるので，ほかの人のバックアップ状況をうかがえる面白い記事でした。私の場合は，なくて困るようなプログラムを複数のディスクに入れ，さらに圧縮をかけて1枚のディスクに収めるようにしています（データに関してはいい加減)。それでも，いままで何度CRCエラーが起きてファイルが読めなくなり，泣かされたことか。私としては，まず，メディアの信頼性向上を望みます。ちょっとしたことですぐに壊れるメディアなどに，いくらバックアップしたところで不安はなくなりませんから。
宍戸　輝光(19)　X68000 PRO,MSX2　東京都

●ショウレポート「データショウ'92」は，なにか寂しい印象を受けました。もっとも時代の流れからして，大容量HDD，広いメモリ，速いCPUのみをうりにしたコンピュータが大半ですから，これはしかたのないことかもしれません。しかし，こんな内容では「人にやさしい情報環境の創造」というサブタイトルが白々しいです。どうせならディスプレイフィルタやキーボードの改良のほうが，よほど「人にやさしい」です。ちらつきや映り込みを防ぐフィルタがあれば目も痛くならないし，キーボードもキータッチによっていろいろ選べてもいいと思います。世の中，ハイテクだ，新素材だ，と騒いでいるのですから，新しい入力装置のひとつも出品してほしいですね。
中村　健(22)　X68000 ACE-HD,AMIGA 500,PC-386GS,MSX2＋　埼玉県

●11月号の「SAVESC.SYS」はよかったですね。昔MSXでコナミの「10倍楽しむカートリッジ」(懐かしいでしょ?)を使って対応ゲームの画面ハードコピーを取って喜んでいた時分を思い出しながら，さっそく打ち込んで遊んでみました。やってみたのは「出たな!!　ツインビー」と「ファイナルファイト」です。一応うまくいきましたが，一度「出たツイ」で敵弾が，「ＦＦ」では（スタート時）キャラが消えてしまったことがありました。それと面白いことに「ＦＦ」でMIDI設定をした状態で画面セーブをすると，終了後にPCMパートなし内蔵音源BGMモード(?)という脅威のモードに入ったこともありました(危ないなぁ)。使ってみて思ったのは，記録するときにその画面をよく覚えておかなくてはならない，ということです。各画面を合わせるときに「出たツイ」なんかでは，ツインビーと敵キャラの位置は正しいと思ってるのに，どうも地上のキノコの位置が悪くて……みたいなことがありました。最近のゲームはグラフィックが凝っていて，見ているだけで楽しめるようなものも多いですから，自分の持っているゲームの名場面集みたいなのを作ってみるのもいいかもしれないですね。
前田　秀樹(19)　X68000 XVI/PRO,MSX,MSX2　京都府

## ごめんなさいのコーナー

**9月号　FPP.MACの作成**

P.90　12月号のごめんなさいのコーナーで，FPP.MACがHAS.Xのバージョンの違いによりアセンブルできない，とありましたがHAS.X ver2.51以降であればアセンブル可能です。

また，旧バージョンでもリスト1中にある“くく”を“.shl.”に置き換えることでアセンブルできるようになります。該当箇所は，126，137，170，179，204，212行です。

**11月号　EDIT**

P.46　テンポラリファイルUPDATE.$$$で出力されるファイル名の区切りが，$0D_H$，$00_H$となることがありました。以下のアドレスに訂正を加えてください。

```
3609  00
3625  22 72 1F 21 00 00 22 70
362D  1F 22 6E 1F CD AF 1F 38
3635  39 2A AB 3A 23
364D  CD C1 3A
3AC1  23 22 72 1F C9
```

**12月号　Oh!X LIVE in'92**

P.125「LAST CHRISTMAS」で，CM-64に設定する音色データが掲載されていませんでした。音色データは，今月号のOh!X LIVE in'93に掲載されていますので，入力方法などはそちらをご覧ください。

**バグに関するお問い合わせは**
**☎03(5488)1311(直通)**
**月～金曜日16：00～18：00**

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。
また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

# ハンダゴテ握って おもちゃで ちゃちゃちゃ

▶今月号の特集では，バーコードリーダー，リモコン制御のバルーン，そしてロボットをX68000で制御してみました。市販されているものを使うためそれほど難しい工作をせず，インタフェイスを確立することによって，動かすことができるようになります。あとは，それらを動かすソフトウェアによっていろいろな遊び方ができるでしょう。興味のある方は，連載のハードウェア工作入門でもいろいろな工作をしてきていますから，読み返してみてあれこれ考えてみてください。

そして，今月から始まった68020アクセラレータボードの製作。記事にあるとおり，現時点の設計段階では，実際にどうなるかまったく予測がつきません。ぜひ，このアクセラレータボードを完成させるためにも，ご意見，情報などをお待ちしています。

▶ところで，ハードウェア特集というと，必ず，手を出したいけどなかなかね……というようなハガキを見ます。やはり，いちばんいけないのは苦手意識をもってしまうこと。ためらっていては先に進めません。

また，純粋にプログラミングだけを楽しんでいる人でも，掘り下げ方によってハードウェアの知識が必要になるでしょう。特にマシンの隅までつつきたい，また，つつく楽しさを味わおうとしたらなおさらです。一歩踏み込んでチャレンジしてみましょう。

▶さて，2月号といえば毎年恒例となっている，GAME OF THE YEARのノミネート発表が行われます。1992年に発売されたゲームの中から，読者投票でベストゲームを選ぶGAME OF THE YEAR。思い入れのあるゲームをアンケートハガキにぶつけてみませんか。

▶そして，3月号では年賀状，クリスマスカードのカラーイラストを紹介する予定です。ページは確保してあるので，あとは読者の皆さんの投稿を待つのみ。頭をひねった楽しいイラスト，気合十分なイラスト，アイデア冴える作品，もちろんCGでも結構です。がんがん投稿してくださいね。

**投稿応募要領**

- 原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。
- プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたテープ（ディスケット）を添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。
- ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討のうえ，製作したハードが必要な場合はご連絡いたします。
- 投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

**あて先**

〒108 東京都港区高輪2-19-13 NS高輪ビル

**ソフトバンク出版部**

Oh!X「テーマ名」係

# SHIFT・BREAK

▶風魔半蔵がほぼリュウケンだろうが，ブロッケンはガイルと趣が違う。ぞうきんがけ男はたまに使うが結局私はドラゴンで空中投げと四段が決まれば幸せなのだ。ジャンヌが最強という話もあるが，個人的にはラスプを推す。奴には勝利の薔薇がよく似合うぜ。そぉれ俺のスカートを食らえぃ！　あぁでもシヌマデパンチはやめてぇ。（30日はヤー38aの哲）

▶追悼スーパーマン。でも寿命はヒーローにしちゃえらく長かったと思う。それに比べると日本のヒーローはやたら古びるのが早い。やっぱりテクノロジーに頼ってるのがいけないんじゃないかな。鉄腕アトムなんか「真空管が切れかかってる」とかいって調子悪くなっちゃうし。そのくせ体内に原子炉を内蔵(！)してるから侮れないよねぇ。（浦）

▶私は前奏曲も好きだが夜想曲のほうが好きだ。ショパンの話である。ホンダのプレリュード(前奏曲)は好みが分かれるところだが，私は気に入っている。あのヒップラインはポルシェに通じるモノがあると思うのだが，いかがだろうか。Z.N.氏が買ったそうなので，一度お手合わせを願いたいところだ。我が家の貴婦人が相手をしたがっている。（S.K.）

▶去年と比べてX68000の年末ゲーム商戦は，意外に地味なものとなってしまったが，映画業界のほうもどーもそんな感じ。「永遠に美しく」「ホームアローン2」も何となく煮えきらないし。やっぱ不景気のせいかな。Z-MUSIC本は筆者多忙により'93年1月下旬から2月初旬になりそう。打楽器以外のAD PCMデータが追加される予定。乞うご期待。（善）

▶「うまく動いている間はいじくらない」がポリシーだった。自分で書いたプログラムは，たとえ少しくらい遅くたって手を入れない。だが体についてはそれはやめたほうがいい。歯って，痛み出したときにはもう手遅れ。診てもらったらもうぼろぼろ。治療がまた痛いし。このときは自分が生命体であることを疎ましく思ってしまった。（冬眠したいA.T.）

▶10年間使っていた旧式電気炊飯器に別れを告げ，最新の「IHでニューロ炊飯でファジイ保温」野郎にリプレースした。しかしまあ，すごい肩書きだよな。確かに，旧式電気炊飯器で炊くよりはおいしいからいいけどさ。図体のわりにカマがちっちゃくて，なんだかなあ，という気がする。というわけで，1年ぶりに自分でご飯を炊いた，という話でした。（K）

▶主人公と2人の子供はみんなレベル99。カジノの成果はメタルキングの剣2本にキラーピアス。エ○○一○は十数ターンで必ず倒せる。さすがに飽きてきたので今度はフローラでやり直している。ビアンカを選ぶとフローラはあっさり結婚したが，今回ではビアンカは結婚しない。みんなの幸せのためにはビアンカを選ぶのが正解なのだろうか。（KO）

▶最近，高校生の頃のことを思い出す機会があった。読者ハガキの影響でCDを聴いてみたのと，渡辺香津美さんにお会いしたことだ。みんな当時好きだった人たちで，渡辺さんの曲は課題のグラフィックでテーマにしたこともある。最近はなぜか少し遠ざかっていたのだが，あの頃の漠然とした夢なんかを思い出した。うん，元気にがんばらなくちゃ！（ふ）

▶この時期になると，頭の中でほしいものリストがぐるぐるとうごめき始める。あれもほしい，これもほしい，と金額を考えずリストアップする作業は結構楽しいものだ。すでに当て込んで買ってしまったものもあるし，思ったとおりのものが出なかったらどうしよう。う〜む，取らぬ狸の皮算用，おきらくごくらく！　の世界だな。（J）

▶立体視は目が疲れる。ちゃんと見えればいいんだけど，見えないときがつらい。何度も焦点を合わせようとして，目がグラグラになっていくのだ。前から落ちていた視力がさらにひどくなったような気もする。メガネを買い換えないといけないかな。でも，これ以上分厚くなると重くて頭を垂れてしまうので，眼球に直接レンズをつけるべきかなあ。（A）

▶みんな冗談だと思っていたら本当に満開製作所を作り，ついに雷語まで作ってしまった。14年もすればマルチプロセッサが容易な64ビットRISC CPUというのは家庭用として妥当なセンだし……。さて，年末多忙のなか入稿を放り出して「どろろ」の全話上映会に行ってきた。半日かけて26話一挙上映。幼い頃に受けた衝撃がようやく確認できた。（U）

▶NICOGRAPH'92に出ていたソニーのソリッドクリエーションシステムは面白い。特殊な液体樹脂の槽に紫外線レーザーを当てると，そこだけ樹脂が固まって，みるみる立体が出来上がっていくのだ。CADデータを元に制御され，複雑な立体も正確に仕上がるという。畳1枚分に満たないスペースでこんなことができるなんてすごい。（T）

## microOdyssey

あのブレードランナーが公開から10年を経たいま，監督リドリー・スコットによる再編集で蘇った。当時は斬新な未来映像で観客を圧倒し，いまもマニアックなファンが多い作品だ。

いったん公開された以上は，あとで手を加えることを認めたくない人もいるだろう。作品の評価，あるいは興行実績を含めた映画の評価に関しては，公開時のものを対象とすべきという人は多い。私も「当初の作品は不本意な点が多かったので，今回のものを最終版として評価してくれ」といわれたらちょっと困ってしまう。もちろん「最終版」というのは日本での興行の際につけられたものだろう。営業上の問題とはいえ，この言い方は結構ムッとくる。

しかし，私にとってのブレードランナーは，このディレクターズカットの登場によっていっそう輝きを増したように思うのだ。変わったのは，主人公デッカードに扮するハリソン・フォードのナレーションがなくなり，デッカードとレイチェルが郊外を逃走するラストシーンもカットされたこと。そのかわりデッカードとレイチェルの感情面を表現するシーンが厚みを増したことだ。デッカード自身もレプリカント（アンドロイド）ではないかと示唆するシーンがあるが，そう考えると旧バージョンで意味不明だった部分も納得がいく。

リドリー・スコットによれば，ナレーションは作品の価値を損ねてしまったが，当時の観客には必要に思ったということだ。確かに，ショッキングな未来映像が観客にとまどいを与え，映像から内容を汲み取ることが難しいという判断は正しかったかもしれない。それが現在の人々は，ブレードランナー的なビジュアルスタイルに慣れてしまったため，もはや過度の説明は必要がなくなったというわけだ。

しかし，もうひとつ重要な観点を忘れてはいけない。スコットは今回のバージョンを，観客が旧バージョンを観ているという前提で編集したのではないかということだ。

私自身，あのナレーションは余計に思えたが，理解を助けられたのも事実だ。そして，ホッとさせられすぎのラストシーンはなくなってよかったと思うが，もしも最初に観たのが今回のバージョンだったとしたら，話の解釈が違っていた可能性が強い。きっと，レイチェルは（あるいはデッカードも）早々に死ぬ運命と解釈していただろう。いやだな，そんなの。

ところで，この映画の原作はフィリップ・K・ディックの『アンドロイドは電気羊の夢を見るか？』だが，ディックの時代の作家たちは，ブレードランナーの観客にこびた楽観性に否定的だった。ひょっとしてナレーションとラストシーンの削除は彼らに対する回答だろうか。たとえそうだとしても，もとからのファンにとっては映像的に洗練されこそすれ，一向に解釈を変える必要はないはずだ。そう考えると，スコットの打った手は実に巧妙といえるだろう。

余談だが，その後リドリー・スコットが撮ったブラックレインで，大阪市内の映像があまりにもブレードランナー的だったのは驚いた。それは確かに私のよく知っている大阪の街であったが，日本映画に出てくる都市の情景とは明らかに違う。なんだか，日本の映画がつまらない理由がわかってしまったようだ。　(T)

## 1993年2月号1月18日(月)発売

**特集　画像創造のために**

・自然画像の自動生成

1992年度GAME OF THE YEARノミネート発表

**詳細レポート**

**X68000用版下作成支援ツールY300-A**

**POLYPHON**

CADモデリングデータコンバータ

## バックナンバー常備店

| | | | |
|---|---|---|---|
| 東京 | 神保町 | 三省堂神田本店5F | 03(3233)3312 |
| | 〃 | 書泉ブックマートB1 | 03(3294)0011 |
| | 〃 | 書泉グランデ5F | 03(3295)0011 |
| | 秋葉原 | T-ZONE 7Fブックゾーン | 03(3257)2660 |
| | 八重洲 | 八重洲ブックセンター3F | 03(3281)1811 |
| | 新宿 | 紀伊国屋書店本店 | 03(3354)0131 |
| | 高田馬場 | 未来堂書店 | 03(3209)0656 |
| | 渋谷 | 大盛堂書店 | 03(3463)0511 |
| | 池袋 | 旭屋書店池袋店 | 03(3986)0311 |
| | 八王子 | くまざわ書店八王子本店 | 0426(25)1201 |
| 神奈川 | 横浜 | 有隣堂横浜駅西口店 | 045(311)6265 |
| | 〃 | 有隣堂ルミネ店 | 045(453)0811 |
| | 藤沢 | 有隣堂藤沢店 | 0466(26)1411 |
| 神奈川 | 厚木 | 有隣堂厚木店 | 0462(23)4111 |
| | 平塚 | 文教堂四の宮店 | 0463(54)2880 |
| 千葉 | 柏 | 新星堂カルチェ5 | 0471(64)8551 |
| | 船橋 | リブロ船橋店 | 0474(25)0111 |
| | 〃 | 芳林堂書店津田沼店 | 0474(78)3737 |
| | 千葉 | 多田屋千葉セントラルプラザ店 | 0472(24)1333 |
| 埼玉 | 川越 | 黒田書店 | 0492(25)3138 |
| | 川口 | 岩渕書店 | 0482(52)2190 |
| 茨城 | 水戸 | 川又書店駅前店 | 0292(31)0102 |
| 大阪 | 北区 | 旭屋書店本店 | 06(313)1191 |
| | 都島区 | 駸々堂京橋店 | 06(353)2413 |
| 京都 | 中京区 | オーム社書店 | 075(221)0280 |
| 愛知 | 名古屋 | 三省堂名古屋店 | 052(562)0077 |
| | 〃 | パソコンΣ上前津店 | 052(251)8334 |
| | 刈谷 | 三洋堂書店刈谷店 | 0566(24)1134 |
| 長野 | 飯田 | 平安堂飯田店 | 0265(24)4545 |
| 新潟 | 新潟 | 紀伊國屋書店新潟店 | 025(241)5281 |
| 北海道 | 室蘭 | 室蘭工業大学生協 | 0143(44)6060 |

## 定期購読のお知らせ

Oh!Xの定期購読をご希望の方は綴じ込みの振替用紙の「申込書」欄にある『新規』『継続』のいずれかに○をつけ，必要事項を明記のうえ，郵便局で購読料をお振り込みください。その際渡される半券は領収書になっていますので，大切に保管してください。なお，すでに定期購読をご利用の方には期限終了の少し前にご通知いたします。継続希望の方は，上記と同じ要領でお申し込みください。

**海外送付ご希望の方へ**

本誌の海外発送代理店，日本IPS(株)にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(3238)0700


1月号
■1993年1月1日発行　定価600円(本体583円)
■発行人　孫　正義
■編集人　橋本五郎
■発売元　ソフトバンク株式会社
■出版事業部　〒108 東京都港区高輪2-19-13　NS高輪ビル
Oh!X編集部　☎03(5488)1309
出版営業部　☎03(5488)1360 FAX 03(5488)1364
広告営業部　☎03(5488)1365
■印　刷　凸版印刷株式会社



# 満開の電子ちゃん

作・え 岡村祭




























講読方法：定期購読もしくはソフトベンダーTAKERUでお買い求めいただけます。
★定期購読の場合＝購読料６ヶ月分6,000円（送料サービス、消費税込）を、
現金書留または郵便振替で下記の宛先へお送り下さい。
現金書留の場合：〒171 東京都豊島区長崎1-28-23 Muse西池袋2F ㈱満開製作所
郵便振替の場合：東京 5－362847 ㈱満開製作所
●ご注文の際は、郵便番号・住所・氏名・電話番号を忘れずに記入して下さい。
●3.5インチディスク版をご希望の方は、「3.5インチ版」とご指定下さい。
●新規購読の方は「新規」と明記して下さい。なお、特に購読開始号のご指定がない場合は既刊の最新号からお送りいたします。
●製品の性格上返品には応じられませんが、お申し出があれば定期購読を解約し残金をお返しします。
★TAKERUでお求めの場合＝１部につき1,200円（消費税込）です。
●定期購読版と内容が一部異なる場合があります。御了承下さい。
●お問い合わせ先 TEL(03)3554－9282（月～金 午前11時～午後６時）
（なお、定期購読版のバックナンバーについては定期購読の方のみご注文を承ります）



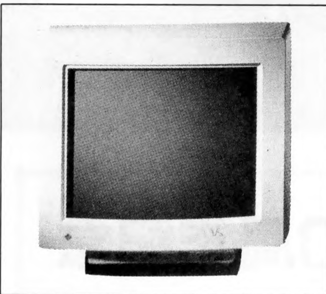
主藤二裕（福岡県）

私と電脳倶楽部との出会いは、このOh!X誌上でした。その時は、ちょうど画像圧縮の特集で、稲妻ツール「PIC・R」が紹介されていたのです。
通信をやっていない私が、それを入手するためには電脳倶楽部が一番てっとり早かったのです。「うそくさいけど試してみるか。」という気持ちで始めた私ですが、ディスクが送られてびっくり！こんなにおもしろくてためになるものだったなんて！
早速バックナンバーを申し込み、いまや私も電倶なしでは生きられないX68kユーザーです。

# 宇都宮新世界発見ワークステーションショップ

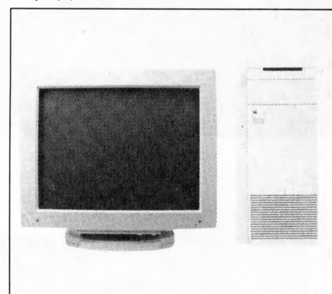
**BASIC HOUSE**
KEISOKUGIKEN Corp.

全国に先駆けてカリフォルニア産の人気マシンを一同に展示中お誘い合わせの上ご来店ください


**SUN SPARC Station**


**IRIS Indigo**


**NeXT Station**


**Apple Macintosh**

# OPEN 12周年記念セール開催中 1月末日まで

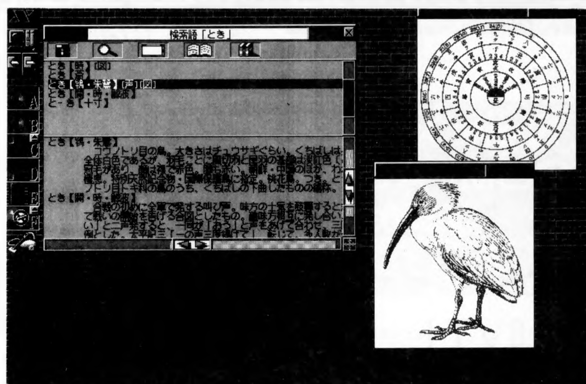
## X-68000用CD-ROM Drive

X-68000, Macintosh
ドライバーソフト
CDキャディ
SCSIケーブル



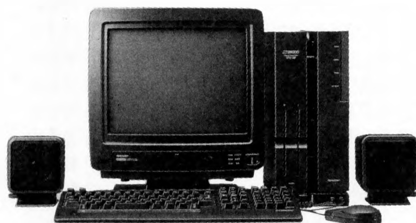
38%OFF
標準価格¥128,000

記念特価　超目玉商品　限定50本限り　特価¥79,800



## X-68000 計測技研オリジナルセット 12周年記念特別価格　限定即納できます!!

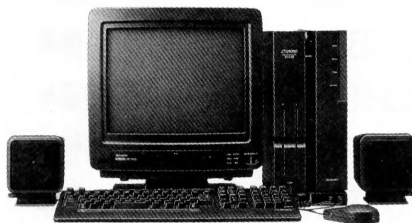
### X68000CompactXVI/HD120
2.5"Quantum120MB内蔵



| | |
|---|---|
| CZ-674C-H | 298,000 |
| Go Drive 120MB | 198,000 |
| 定価合計 | 496,000 |

**36%OFF　318,000**

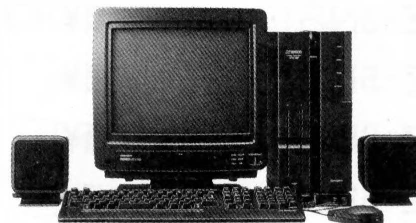
### X68000XVI/HD240
3.5"Quantum240MB内蔵



| | |
|---|---|
| CZ-634C-TN | 368,000 |
| Quantum 240MB | 198,000 |
| 定価合計 | 496,000 |

**36%OFF　368,000**

### X68000XVI/HD425
3.5"Quantum425MB内蔵



| | |
|---|---|
| CZ-634C-TN | 368,000 |
| Quantum 425MB | 298,000 |
| 定価合計 | 496,000 |

**33%OFF　448,000**

12月14日新発売 定価
X68000 CD-ROM第一弾!!
**FREE SOFTWARE SELECTION**
中味は買ってからのお楽しみとにかくすごいものがたくさん入っています

超目玉 3台限り
CZ-8PC5-BK 熱転写カラー漢字プリンタ

定価96,800円　49,800円　50%OFF

# X68000 PROSHOP

## 株式会社計測技研

本社ショールーム　〒321 栃木県宇都宮市竹林町503-1
TEL0286-22-9811　FAX0286-25-3970
研究開発部門　First Class Technology
〒320 栃木県宇都宮市京町11-18
TEL0286-38-0301　FAX0286-38-0305
Sunnyvale営業所　875 Cumbertand Drive Sunnyvale, CA. 94087
TEL408-720-1573　FAX408-720-1576

# マイコンショップ川口

☎048-225-1718

(消費税別)



New X68000
COMPACT XVI
~~¥298,000~~

| | |
|---|---|
| CZ-674C-H | 定価¥298,000 |
| CZ-608D-H | 定価¥ 94,800 |
| AV-090-SC | 定価¥168,000 |

**超特価 TELTさい**

| | |
|---|---|
| CZ-634C-TN | ㊤ 368,000 |
| CZ-644C-TN | ㊤ 518,000 |

**ソフト各種超特価ご奉仕中**

| | |
|---|---|
| CZ-219SS OS-9/X68000 | 定価¥29,800 |
| CZ-213MS MUSIC PRO68K | 定価¥18,800 |
| CZ-214MS SOUND PRO68K | 定価¥15,800 |
| CZ-215MS Sampling PRO68K | 定価¥17,800 |
| CZ-220BS DATA PRO68K | 定価¥58,000 |
| CZ-224LS The 福袋 Ver2.0 | 定価¥ 9,980 |
| CZ-225BS Multiword | 定価¥32,000 |
| CZ-251BS Hyper word | 定価¥39,800 |

## 開店10周年記念

大奉仕キャンペーン実施中!!

| 品名 | 定価 | 売価 |
|---|---|---|
| CZ-674C-H本体 | ¥298,000 | 大特価 |
| CZ-634C-TN本体 | ¥368,000 | 大特価 |
| CZ-644C-TN本体 | ¥518,000 | 大特価 |
| CZ-608D-Hディスプレイ | ¥ 94,800 | 大特価 |
| CZ-606D-TNディスプレイ | ¥ 79,800 | 大特価 |
| CZ-607D-TNディスプレイ | ¥ 99,800 | 大特価 |
| CZ-614D-TNディスプレイ | ¥135,000 | 大特価 |

## プリンター

| | |
|---|---|
| CZ-6VT1 | 特価¥ 47,700 |
| CZ-8PG1 | 特価¥ 86,800 |
| CZ-8PG2 | 特価¥106,900 |
| CZ-8PK10 | 特価¥ 66,800 |
| CZ-8NS1 | 特価¥141,000 |
| CZ-6BC1 | 特価¥ |
| CZ-6BG1 | 特価¥ |
| CZ-6BP1 | 特価¥ |
| CZ-6BP2 | 特価¥ 34,400 |

## ラムボード

| | | |
|---|---|---|
| CZ-6BE2A | 定価¥59,800 | 特価¥ 44,900 |
| CZ-6BE2B | 定価¥54,800 | 特価¥ 41,100 |
| CZ-6BE2D | 定価¥54,800 | 特価¥ 41,100 |
| CZ-6BE1B | 定価¥28,000 | 特価¥ 21,000 |
| CZ-6BE2 | 定価¥79,800 | 特価¥ |
| CZ-6BE4C | 定価¥98,000 | 特価¥ |
| PIO-6BE1-A | 定価¥25,000 | 特価¥ |
| PIO-6BE2-2M | 定価¥50,000 | 特価¥ |
| PIO-6BE4-4M | 定価¥88,000 | 特価¥ |
| SH-6BE1-1M | 定価¥25,000 | 特価¥ |

## ファイル

| | | |
|---|---|---|
| CZ-6MO1 | 定価¥450,000 | 特価¥ |
| CZ-64H | 定価¥120,000 | 特価¥ |
| CZ-68H | 定価¥160,000 | 特価¥ |

## その他機種

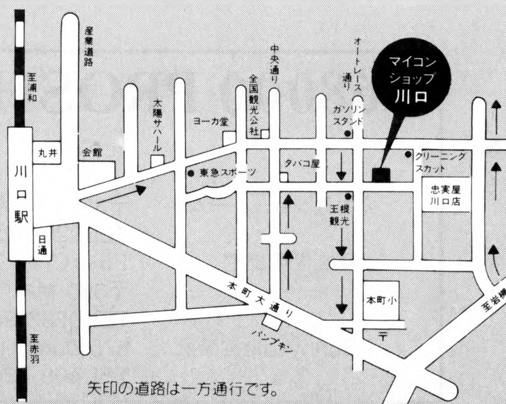
| | | | |
|---|---|---|---|
| CZ-8NS1 | カラーイメージスキャナ | 定価¥188,000 | 特価¥ |
| JX-220X | カラーイメージスキャナ | 定価¥168,000 | 特価¥ |
| CZ-6BN1 | スキャナ用パラレルボード | 定価¥ 29,800 | 特価¥ |
| CZ-6VT1 | カラーイメージユニット | 定価¥ 69,800 | 特価¥ |
| CZ-6BV1 | ビデオボード | 定価¥ 21,000 | 特価¥ |
| CZ-8TM2 | モデムユニット | 定価¥ 49,800 | 特価¥ |
| CZ-8NJ2 | インテリジェントコントローラ | 定価¥ 23,800 | 特価¥ |
| CZ-8NM3 | マウス・トラックボール | 定価¥ 9,800 | 特価¥ |
| CZ-8NT1 | トラックボール | 定価¥ 6,888 | 特価¥ |
| CZ-8NJ1 | ジョイカード | 定価¥ 1,700 | 特価¥ |
| CZ-6BC1 | FAXボード | 定価¥ 79,800 | 特価¥ |
| CZ-6BM1A | MIDIボード | 定価¥ 26,800 | 特価¥ |
| CZ-6BP1 | 数値演算プロセッサ | 定価¥ 79,800 | 特価¥ |
| CZ-6BP2 | 数値演算プロセッサ | 定価¥ 45,800 | 特価¥ |
| CZ-6TU-BK-GY | RGBシステムチューナー | 定価¥ 33,100 | 特価¥ |

★クレジット回数1〜60回まで設定自由

| 回数 | 1 | 3 | 6 | 12 | 15 | 20 | 24 | 36 | 42 | 48 | 54 | 60 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 金利(%) | 2.5 | 2.9 | 3.9 | 5.4 | 8.4 | 10.9 | 11.4 | 15.9 | 19.9 | 20.9 | 25.9 | 26.9 |

**中古品も取扱っております。**




## 通信販売をご利用の方

—全国通販—

品名及びお客様の住所・氏名・電話番号をお知らせ下さい。



▲上図は、図形・文字ともY300－Aで作成し、CZ－8PC5で印刷した鹿児島県の地図を約35％に縮小したものです。

# フィニッシュワークの強い味方、新登場！

■版下作成支援プログラム **Y300-A** ▷ 豊富な機能を満載して **29,800円**

**マウスによる簡単操作と多彩な機能で、あなたの版下制作作業を支援します。**

- Y300－Aで使用する単位はミリで、版下の作成から出力まで、すべて実際の寸法で行います。
- 版下の大きさは、横・縦ともに5ミリから2,000ミリまで、自由に指定できます。
- 書体倶楽部のアウトラインフォントに対応。
  Y300－Aで扱う文字は、すべてアウトラインフォントのため、附属の単線文字〔半角・全角（非漢字、JIS第一水準）〕か、書体倶楽部のフォントをご利用ください。（『書体倶楽部』は、株式会社Zeitの商標です。）
- 版下作成に必要なトンボも３種類用意いたしました。
  トンボデータは自動的に付加されるので、煩わしさがありません。
- 作成したデータを有効に利用するシンボル機能。
- 最大８階層のレベル機能。
- スキャナーで地図・マーク等を取り込み、トレースすることができます。
- 一枚の版下に記入できる図形および図形文字の数は、ディスクの空き容量によって変わります。また、文章用として記入できる文字数は約5,900字です。
- フロッピーシステムで使用する時は漢字は使用できません。

対応機種……X68000（要2M以上）
Human68K Ver.2.0以降が必要です。

〔対応プリンタ〕SHARP CZ系（24ピン・48ピン）
Canon BJ－10V
NEC PC－PR201
EPSON ESC／P24－84J
〔対応プロッタ〕Roland DXY1000シリーズ
GRAPHTEC MP4000シリーズ
〔対応スキャナ〕OMRON HS7R
HAL HGS68附属のソフトで作成した拡張ベタファイル

※カラー印刷はできません。

**全国通販受付中！**

Y300－Aは、通信販売でお求めください。

住所・氏名・電話番号を明記の上、代金29,800円（税込み・送料サービス）を現金書留または銀行振込みにてお送りください。釣り銭のいらないようにお願いいたします。なお、銀行振込みの場合は、事前に住所・氏名・電話番号をお知らせください。

お申し込み・お問い合わせは、

**マグマソフト**
〒891-01 鹿児島市東谷山三丁目32-29
TEL（0992）68-2286
〈銀行振込み先〉南日本銀行東谷山支店 普通357169

# X68000ユーザー必読書!!




## Inside X68000

桑野雅彦 著

画面制御関連はもちろん、LSIについても詳説。GCCによるサンプルプログラム付。

定価6,800円




## GNUツールボックス

吉野智興・村上敬一郎 著

X68000上でのCプログラム作成について、初歩からわかりやすく解説。

定価2,600円




## X68000 C プログラミング

中森章 著

GNUをX68000に移植するためのノウハウについて解説。定価2,200円




## SX-WINDOW プログラミング

吉沢正敏 著

内部解説にもとづいたプログラミングの実例を解説。定価4,500円

**追補版**一定価4,200円
[Ver.1.10対応/ディスク付]



## X68000マシン語プログラミング

〈入門編〉〈グラフィック編〉

村田敏幸 著

豊富な実例とともにマシン語プログラミングの面白さを解説。

**入門編**一定価2,800円
**グラフィック編**一定価3,600円[ディスク付]

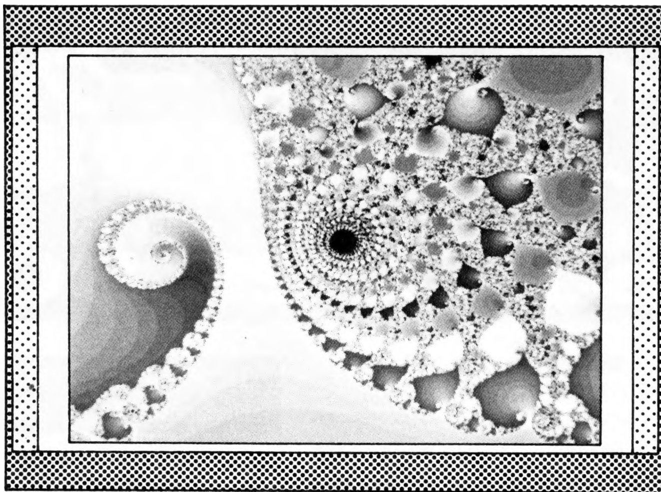
定価は税込みです

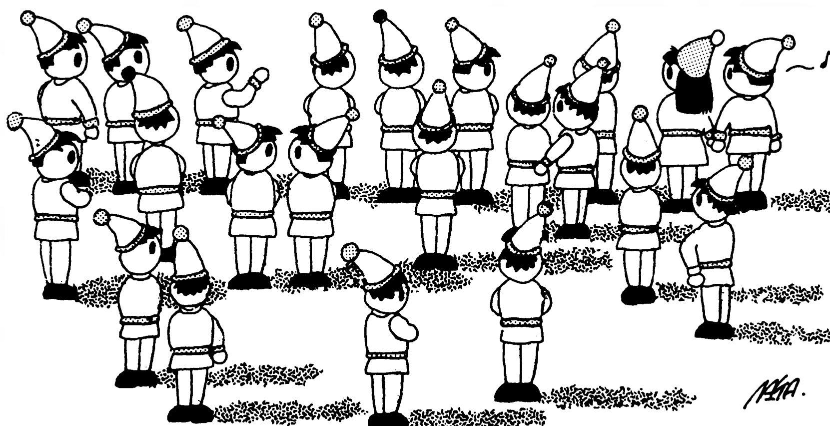
ソフトバンク株式会社/出版事業部

ACCESS

for X68000

# V70アクセラレータならすぐに展覧会が開けます。





V70アクセラレータが数値演算で高いパフォーマンスを誇るのは、クロック20MHzのV70CPUを搭載し、さらにAFPP(フローティング・ポイント・プロセッサ)を標準装備しているからです。特にコンピュータ・グラフィックスの世界では、その実力を十二分に発揮することができるでしょう。写真のグラフィックスでは、実行速度で約45倍のパフォーマンスを記録しました。開発環境に関しても、アセンブラ、リンカはもちろん、ソースコードデバッガやフロートエミュレータ・コマンドシェルと、V70の特徴である仮想記憶、メモリプロテクション等をサポートする充実した開発環境が整っています。V70アクセラレータは、一所懸命に作ったプログラムの実行結果をすぐに見たい／というあなたの願いを、きっとかなえてくれるボードです。

```
for(x=0;x<512;x++) {
        for(y=0;y<512;y++) {
            X=0.0;
            Y=0.0;
            for(t=1;;t++) {
                if(t==512) {
                    break;
                }
                Q=X*X-Y*Y+x*T+p[0];
                R=2*X*Y+y*U+p[2];
                if((Q*Q+R*R)>4.0) {
                    break;
                }
                X=Q;
                Y=R;
            }
            psetptr.x=x;
            psetptr.y=y;
            psetptr.color=((t)%256);
            PSET(&psetptr);
            palat[y]=(unsigned char)psetptr.color;
        }
    }
}
```

上記グラフィックス(フラクタル)作成の為のプログラム(主要演算部分)

**上記グラフィックスの描画速度比較**

X68000(10MHz+FPP無し)+FLOAT2.X……約27時間10分
X68000(10MHz)+VDTK-X68K……………………約37分！

## VDTK-X68Kの仕様

- V70 CPU(μPD70632) 20MHz 32ビットマイクロプロセッサ
- V70 AFPP(μPD72691) フローティング・ポイント・プロセッサ
- メインメモリ(DRAM)2Mバイト 同一ページ内のアクセスはNo Wait
- 共有メモリ(SRAM)128Kバイト X68000との通信用
- 併行動作 X68000とV70は、併行に動作することが可能。データの受け渡し処理のために双方向ハンドシェークI/Oポートを搭載。

## 同梱ソフトウェア

- アセンブラ
- リンカ
- ソースコードデバッガ
- システムモニタ
- フロートエミュレータ
- コマンドシェル

## 価格

- ボードパッケージ(XVI対応)
  **VDTK-X68K …………¥248,000**
- オプションソフト(Cコンパイラ)
  **VDTK-C-X68K………¥68,000**

## オプションソフトウェア

- Cコンパイラ (VDTK-C-X68K)

## 購入方法

上記商品は当面の間、通信販売のみとさせて頂きます。購入ご希望の方は、住所、(社名、所属)氏名、電話番号をお知らせ下さい。注文書をお送りいたします。

※製作:ボード……有限会社アクセス
ソフトウェア……株式会社ハドソン

有限会社アクセス 〒101 東京都千代田区神田神保町1-64 神保町協和ビル7F
☎03(3233)0200(代) FAX.03(3291)7019

資料請求券 oh!X 1月号

パソコン/ワープロ通信ネットワークサービス

# J&P HOT LINE

タイムトラベルシリーズ

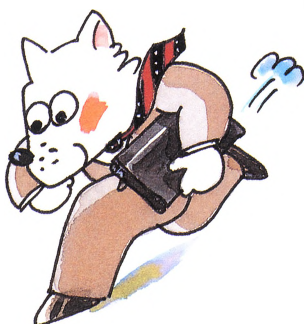
第10回

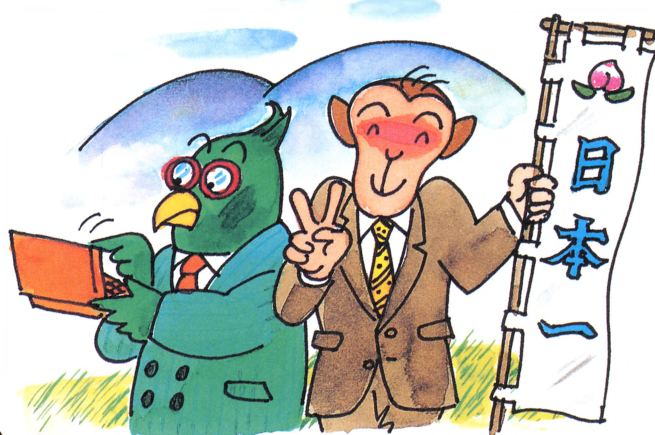
《もし、この時代にパソコン通信があったなら》

## 取り引きの名手

### 桃太郎

ある川に流れついた大きな桃。
その地で暮らすおじいさんとおばあさんが、
その桃を割ってみると、
中から出て来たのは男の赤ちゃんでした。
その子は桃太郎と名づけられ、
大きくなってから、
鬼ヶ島に鬼退治に出かけました。
途中、おばあさんからもらった
きび団子と引き換えに、
イヌ・サル・キジを仲間にし、
見事な連携プレーで
鬼たちを退治してしまったのでした。



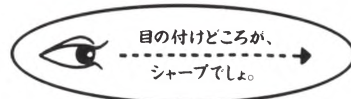



**もし、この時代にパソコン通信があったなら…**

もし、この時代にパソコン通信があったなら。桃太郎は優秀なビジネスマンになっていたかもしれません。なにせ、きび団子という身近な持ち物だけで、プロジェクトチームを組む凄腕です。パソコン通信なら、自分が持つ身近な情報こそが宝物。独特の視点の情報を提供することで、何十、何百の仲間を見つけ出していたことでしょう。

**パソコン通信なら、こんな楽しさ。**

パソコン通信は、互いに情報を提供しあえるところが魅力。あなたが持つなんでもない情報が、他の人には宝物のように重要な情報になることもあります。気楽でメリットも大きいギブアンドテイク。パソコン通信なら、より良いコミュニケーションが図れます。








# いわば“感性”専用。

ことマインドに関しては
「汎用」という概念は存在しないも同じです。
「実用的である」のと、これなら「使える」というのも違います。
X68000が、普通のパソコンとは違うといわれる所以もここにあります。
いわゆる実用性を重視したビジネスパソコンとは
創造力で一線を画しています。
何に使うのか、何がしたいのか、
パソコン選びのポイントは目的にあったマシンを探すこと。
普通のパソコンに合わせるのでは
あなたのせっかくの創造力も発揮されません。
X68000は、使う人のクリエイティブマインドを咲かせる
“感性”専用パソコンです。

## X68000
PERSONAL WORKSTATION・XVI
## Compact

**本体＋キーボード＋マウス**
2HD3.5インチFDDタイプ CZ-674C-H（グレー） 標準価格298,000円（税別）
**14型カラーディスプレイ（ドットピッチ0.28mm）**
CZ-608D-H（グレー） 標準価格94,800円（税別）

- 5.25インチ増設用フロッピーディスクドライブ CZ-6FD5 標準価格99,800円・税別〔接続ケーブル同梱〕
- ディスプレイテレビ/CZ-6TU用RGBケーブル CZ-6CR1 標準価格4,500円・税別
- ディスプレイテレビ/CZ-6TU用テレビコントロールケーブル CZ-6CT1 標準価格5,500円・税別
- SCSI変換ケーブル CZ-6CS1 標準価格12,000円・税別




（カラー液晶ディスプレイとの組み合わせ例）
10.4型TFTカラー液晶ディスプレイ
LC-10C1-H（グレー）標準価格598,000円（税別）
接続ケーブル AN-1515X 標準価格4,200円（税別）

※カラー液晶ディスプレイを接続してご使用の場合、SX-WINDOW上のアプリケーション利用に限定されます。

●お問い合わせは…
シャープ株式会社 電子機器事業本部システム機器営業部〒545大阪市阿倍野区長池町22番22号☎(06)621-1221（大代表） 電子機器事業本部AVCシステム事業推進室〒162東京都新宿区市谷八幡町8番地☎(03)3260-1161（大代表）



T1002179010605 雑誌 02179-1

1月号 1993年1月1日発行（毎月1回1日発行）通巻129号 昭和58年11月2日第三種郵便物認可 ソフトバンク株式会社発行 Printed in Japan 定価600円（本体583円）